

UNIVERSIDAD POLITÉCNICA DE MADRID
Escuela Técnica Superior de Ingenieros Industriales
Departamento de Automática, Ingeniería Electrónica e Informática Industrial

DETECCIÓN E IDENTIFICACIÓN DE OBJETOS MÓVILES EN SISTEMAS MULTI-ROBOT

Gonzalo Ruy Rodríguez Canosa

Proyecto de Fin de Máster – Máster de Automática y Robótica

Dirigido por

Jaime del Cerro Giner y Antonio Barrientos Cruz

15 de Septiembre de 2010

I. AGRADECIMIENTOS

Quisiera agradecer en primer lugar la ayuda prestada por mis dos tutores. A Antonio por lo buenos consejos al empezar el máster y la ayuda al llegar a Madrid y a Jaime, por todo el trabajo diario y las indicaciones para salir de

II. ÍNDICE

I. AGRADECIMIENTOS	1
II. ÍNDICE	2
III. ÍNDICE DE FIGURAS.....	4
IV. RESUMEN.....	5
V. ABSTRACT	6
1. INTRODUCCIÓN.....	7
1.1. Motivación y marco del Proyecto de Fin de Máster. Robots y sistemas multirobot de seguridad.....	7
1.2. Objetivos del Proyecto Fin de Máster.	9
1.3. Estructura del documento	10
2. ESTADO DEL ARTE.....	12
2.1. Reconocimiento de objetos dinámicos.....	12
2.2. Reconocimientos de objetos dinámicos mediante telemetría Láser.	14
2.2.1. Escáneres Láser.....	15
2.2.2. Técnicas de análisis aplicadas.....	17
2.3. Seguimiento de objetos	20
2.3.1. Técnicas utilizadas en sistemas de un solo robot.....	20
2.3.2. Técnicas utilizadas en sistemas multirobot	22
2.4. Información utilizada para la identificación de objetos dinámicos.....	23
2.4.1. Datos del láser. Matrices de distancia.	23
2.4.2. Datos de la cámara y la cámara infrarroja. Extracción del color y la temperatura.	24
3. ALGORITMOS DE DETECCIÓN DE OBJETOS MÓVILES	25
3.1. Detección de movimiento.....	25
3.2. Creación y actualización de la matriz de referencia.	27
3.3. Matrices de colisión.....	28
3.4. Extracción de características reales.....	32
4. ALGORITMOS DE IDENTIFICACIÓN.....	34
4.1. Algoritmo de identificación por mínima distancia.	35
4.2. Algoritmo de filtro de Kalman de estado variable.....	39
4.2.1. Iteración del filtro de Kalman. Adición y extracción de objetos.....	43

5. SISTEMAS MULTIROBOT. FUSIÓN DE INFORMACIÓN PARA LA IDENTIFICACIÓN DE OBJETOS DINÁMICOS.....	46
5.1. Algoritmo de fusión de información	48
5.2. Arquitecturas utilizadas. Ventajas e inconvenientes.....	51
6. DETECCIÓN DE OBJETOS MÓVILES DESDE UNA PLATAFORMA EN MOVIMIENTO.....	53
7. APLICACIÓN AL PROYECTO NM-RS. SIMULACIONES REALIZADAS.....	59
7.1. Detección de objetos desde robot estático.	59
7.1.1. Detección de un objeto en movimiento.....	59
7.1.2. Detección con oclusiones. Dos objetos en movimiento.	61
7.1.3. Prueba con 3 objetos	63
7.2. Detección e identificación con el algoritmo de filtro de Kalman	65
7.2.1. Detección e identificación de 3 objetos.....	65
7.3. Fusión de información entre varios robots	68
7.4. Detección de movimiento desde robot dinámico.....	68
7.4.1. Dos robots enfrentados	68
7.4.2. Dos robots en paralelo.....	70
8. LOGROS, CONCLUSIONES Y DESARROLLOS FUTUROS.	73
9. BIBLIOGRAFÍA.....	76

III. ÍNDICE DE FIGURAS

Figura 1: Esquema de un sensor LIDAR.....	16
Figura 2: Ejemplo de mapa de celdas.....	18
Figura 3: Representación por colores de una matriz de distancias	24
Figura 4: Ejemplo de diferencia de matrices de distancias	25
Figura 5: Ejemplo de matriz tras el proceso de filtrado.....	26
Figura 6: Sistemas de referencia empleados en el robot.....	30
Figura 7: Esquema para la obtención de características reales de los objetos dinámicos	33
Figura 8: Esquema del algoritmo de identificación de objetos.....	38
Figura 9: Esquema del algoritmo de fusión de información	50
Figura 10: Esquema del flujo de información para estructura centralizada	51
Figura 11: Esquema del flujo de información para estructura descentralizada	52
Figura 12: Ejemplo de mapa de puntos de colisión	55
Figura 13: Ejemplo de desplazamiento del mapa a medida que se mueve el robot	56
Figura 14: Esquema del algoritmo de extracción de movimiento	57
Figura 15: Ejemplo de obtención de mapa de movimiento	58
Figura 16: Disposición inicial de los robots	59
Figura 17: Resultados de la prueba con un sólo robot dinámico.....	60
Figura 18: Posiciones iniciales de los robots	61
Figura 19: Resultados de la prueba con dos robots dinámicos.....	62
Figura 20: Posición inicial y trayectorias de los robots	63
Figura 21: Resultados de la prueba con 3 robots.....	64
Figura 22: Resultados de la identificación con el algoritmo de filtro de Kalman.....	66
Figura 23: Resultados superpuestos de los dos algoritmos para la prueba con 3 objetos dinámicos	67
Figura 24: Posición inicial de los robots en la prueba	68
Figura 25: Resultados de la prueba de dos robots enfrentados	69
Figura 26: Posición inicial de los robots en la prueba	71
Figura 27: Resultados de la prueba de los dos robots paralelos.....	72

IV. RESUMEN

El presente documento contiene la memoria del proyecto de fin de máster titulado *Detección e identificación de objetos móviles en sistemas multirobot*. El principal objetivo de este proyecto es desarrollar una serie de algoritmos que permitan detectar e identificar objetos móviles desde una plataforma robótica, así como compartir la información obtenida entre los diferentes robots que componen el sistema.

En primer lugar se ha realizado un estudio de la necesidad de estos sistemas y del estado actual de la tecnología en este campo, aplicado en su mayor parte a la seguridad y vigilancia. También se ha realizado un estudio en profundidad del estado del arte en estos sectores, en particular el análisis de sistemas láser y algoritmos de seguimiento de objetos dinámicos (*DATMO – Detection and Tracking of Mobile Objects*).

Una vez estudiado el problema y las soluciones ya existentes se ha optado por desarrollar una solución teórica aplicable a diferentes plataformas y sistemas. Se ha separado el problema en tareas independientes para encontrar la mejor solución a cada una: la detección, la identificación y la fusión de información entre las diferentes plataformas.

Para la detección (véase capítulo 3), se ha optado por la diferenciación directa de las matrices de distancia del láser, aplicando después una serie de filtros para obtener los puntos que conforman cada objeto. Se han utilizado dos algoritmos de identificación diferentes (véase capítulo 4). Mientras que uno de ellos utiliza solamente la información obtenida del láser, el otro fusiona la información proveniente de varios sensores y, mediante un filtro de Kalman de estado variable, predice e identifica los objetos de forma óptima. En el capítulo 5 se describe un algoritmo de fusión de información que permite integrar las listas de objetos detectadas por diferentes robots, compartiendo así la información entre ellos.

Como este sistema pretende estar instalado en robots móviles se ha desarrollado un algoritmo de detección de movimiento a partir del láser que funciona mientras el robot se desplaza (véase capítulo 6).

Por último, se han aplicado todos estos desarrollos a un proyecto europeo, el NM-RS (*Networked Multi-Robot Systems*), incluyéndose en el capítulo 7 los resultados de algunas de las simulaciones realizadas. Los resultados obtenidos han sido satisfactorios y se ha comprobado que todos los algoritmos cumplen su propósito y son útiles a la hora de desempeñar la función para la que han sido desarrollados.

V. ABSTRACT

The present document contains the Final Project of the Master thesis entitled *Detection and Identification of Mobile Objects in Multi-Robot Systems*. The main objective of this project is to develop a group of algorithms that enable the detection and tracking of mobile objects from a robotic platform, as well as to share the information between all the robots that belong to the system.

Primarily, we have studied the necessity of these systems, along with the current state of the technology in this field, applied mostly to security and surveillance functions. A thorough study of the state of the art in these sectors has been also carried out, paying particular attention to the analysis of laser systems in relation with detection and tracking of mobile objects (DATMO) algorithms.

Once this problem and the preexistent solutions have been studied, we have selected a theoretical approach able to be applied to a large range of platforms and systems. The problem has been separated in three sections, to find the better solution for each one: detection, identification and merging of the information among the different platforms.

For the detection (see chapter 3), we have chosen the direct differentiation of the distance matrixes from the laser, together with a series of filters to obtain all the points belonging to an object. Two different identification algorithms have been developed (see chapter 4). While one of them uses only the information coming from the laser, the other one fuses the information coming from different sensors and with a Kalman filter of variable state, predicts and identifies the objects in an optimal way. In chapter 5 a merging algorithm is described, that enables the different robots to combine the objects lists detected by each other by sharing the information between them.

Since the system intends to be installed in mobile robots, a movement detection algorithm has been developed that allows the system to determine in which areas of the environment there are moving objects (see chapter 6).

Finally, all these developments have been applied to an European project, NM-RS (*Networked Multi-Robot Systems*). The results of some of the performed simulations can be seen in chapter 7. The obtained results are satisfactory and it has been proven that the algorithms are useful for the intended application and can be used in a real working environment.

1. INTRODUCCIÓN

1.1. Motivación y marco del Proyecto de Fin de Máster. Robots y sistemas multirobot de seguridad.

La vigilancia de grandes estructuras es un aspecto muy importante de la seguridad de las infraestructuras críticas de cualquier estado o empresa. Por esa razón se está impulsando cada vez más el desarrollo de soluciones robóticas para este tipo de misiones, tanto dentro del ámbito civil como el militar. Centrales de energía, centros de distribución de alimentos, infraestructuras de transportes e instalaciones de almacenamiento o producción de materiales peligrosos (químicos, biológicos o nucleares) se encuentran dentro de la lista de posibles emplazamientos de este tipo de desarrollos tecnológicos.

Los desarrollos comerciales actuales de soluciones robotizadas para sistemas de vigilancia y seguridad en el ámbito civil han experimentado grandes mejoras en los últimos años [1, 2, 3, 4]. Sin embargo todavía presentan problemas de importancia en algunas áreas vitales para la implantación masiva de estos sistemas. Los principales problemas se presentan en áreas como la localización del robot, ya que el GPS diferencial sigue siendo muy costoso y no se puede usar en interiores; la locomoción del robot en diferentes tipos de terrenos; y la localización y navegación en entornos exteriores.

Otra de las áreas donde es necesaria una amplia investigación y desarrollo de nuevas tecnologías y procedimientos es la detección e identificación de objetos móviles dentro del perímetro de seguridad. Si bien existen numerosos sensores que se pueden utilizar para esta labor (sistemas láser, cámaras infrarrojas, sistemas de ultrasonidos o cámaras de espectro visible), todavía no se emplea toda esta información para una correcta detección e identificación. En la mayoría de los casos, una vez se detecta movimiento la mayor capacidad de las aplicaciones comerciales consiste en enviar una imagen a un operador remoto para que él decida el nivel de la amenaza que supone cada intrusión. Un sistema de actuación ante detección de intrusos completamente autónomo es difícil de realizar; siendo más factible un sistema híbrido que incluya un operador y una red de sensores y robots semiautónomos.

El problema del reconocimiento de intrusos no se solventa simplemente mejorando los sistemas de detección e identificación de esos objetos móviles; también es necesario poder coordinar una respuesta ante esta amenaza. Para la correcta implementación de un sistema robótico de seguridad es necesaria la integración de

todos los robots, junto con sensores de diferente tipo e incluso algún supervisor humano, en un sistema híbrido que pueda funcionar de forma autónoma o semi-autónoma combinando información de diferentes fuentes.

En este marco de investigación se puede destacar el proyecto NM-RS de la Agencia Europea de Defensa (NMRS / B-0004-ESM2_ERG [5]). El principal objetivo del proyecto es desarrollar un demostrador virtual que permita la planificación y demostración de operaciones de un sistema multirobot en red, tanto en ambientes urbanos como no urbanos. Este proyecto se enmarca dentro de la categoría B y se encuentra dirigido por una empresa alemana (*DBD, Diehl BGT Defense*) con socios belgas, españoles e italianos; tanto procedentes del sector empresarial como del mundo académico. El proyecto pretende evaluar las ventajas que tiene el uso de sistemas en red frente a robots individuales o al empleo de tropas en aplicaciones de defensa. Con este objetivo se desarrollarán tácticas y procedimientos para el empleo de estos sistemas robóticos en labores de seguridad y defensa, durante periodos mucho más largos que las otras alternativas. Las capacidades de la plataforma a desarrollar serán varias, destacando la posibilidad de simular sistemas de varios robots que se mueven de forma autónoma en entornos estructurados y no estructurados, con la supervisión de un operador que recibe la información generada por todos los robots.

Otro de los logros que se pretenden alcanzar con esta plataforma es proporcionar al personal militar la posibilidad de entrenarse en el uso de estos sistemas robóticos sin tener que recurrir a costosos equipos y robots, así como evitando las averías. Esto permitiría además experimentar con las interfaces de operación para determinar el número de robots que una tripulación puede controlar en diferentes escenarios; ya que se pretende que el software desarrollado sea usado en un futuro para controlar grupos de robot trabajando en equipo.

También se han desarrollado algoritmos multi-robot de aplicación en tareas colaborativas bajo diferentes niveles de comunicación entre los robots. Estas técnicas innovadoras permitirán a los robots construir memorias de sus experiencias sobre el entorno, evaluar la utilidad de las alternativas de acciones cooperativas y entonces seleccionar acciones a realizar de tal forma que se incremente la probabilidad de que el objetivo global se cumpla, más allá de las decisiones individuales de cada robot [6].

Tras desarrollar los algoritmos y soluciones presentadas en esta Tesis de Máster, se han aplicado a este proyecto dentro del módulo de detección de objetos dinámicos.

1.2. Objetivos del Proyecto Fin de Máster.

Este proyecto encara dos aspectos diferentes de la detección e identificación de diferentes objetos móviles aplicada a un sistema multi-robot: la detección e identificación de objetos móviles y la fusión de información entre diferentes robots dispuestos en dos configuraciones (centralizado y descentralizado).

En primer lugar se pretende desarrollar un método de detección de objetos móviles basado en las medidas de un sistema láser en 3D, para después combinar esta información con la de otros sensores (como una cámara termográfica y una de espectro visible). Esta fusión sensorial permitirá realizar un seguimiento de la trayectoria y características de un determinado objeto facilitando así la labor de evaluación del nivel de amenaza.

El segundo aspecto que se pretende resolver con este proyecto es la fusión de información entre diferentes robots permitiendo así al sistema un mayor control sobre los objetos móviles dentro de la zona de seguridad. Se pretende estudiar la fusión de información entre los robots tanto de forma centralizada como descentralizada. En el primer caso existirá una base central que conecte con todos los robots, recibiendo, procesando y enviando de vuelta toda la información sobre los objetos móviles. Para el caso descentralizado se elimina la existencia de la base central y son los propios robots los que se envían la información unos a otros añadiendo a sus propias listas, la información de objetos dinámicos recabada por otros robots. En ambos casos, aunque de forma diferente, los robots pueden tener conocimiento de la existencia de un objeto móvil antes de percibirlo con sus sensores.

Una vez descrito los objetivos generales del proyecto, hay que enumerar los objetivos particulares de cada una de las dos partes.

- Creación de una lista de objetos dinámicos a partir de las medidas tomadas por el láser.
- Extracción del resto de las características, en concreto color y temperatura, de los demás sensores y su posterior incorporación a cada uno de los objetos.
- Utilización de todas las características de los objetos encontrados en cada iteración para asociarlos con los de la iteración anterior (en el caso de ser posible esa asociación) y crear una nueva lista.
- Fusión de las listas generadas por cada robot.
 - o De forma centralizada, en una estación de control que se comunica con todos los robots.

- De forma descentralizada, entre los propios robots, intercambiando información cuando se encuentran dentro del alcance de su sistema de comunicación.
- Probar todos los algoritmos y procedimientos desarrollados dentro del marco del proyecto NM-RS y analizar los resultados para demostrar la viabilidad de los métodos.

Todo el trabajo que se pretende desarrollar en este proyecto es para entornos 3D, con las particularidades que ello conlleva. En primer lugar, hay que procesar una cantidad de información mucho más grande que si se utilizan sensores 2D, lo que obliga a que los algoritmos estén optimizados para que se puedan ejecutar en tiempo real. También hay que tener en cuenta que la información que se obtiene de este tipo de sensores y algoritmos es mucho más completa de la que se puede obtener con sensores 2D aunque se utilicen varios sensores.

1.3. Estructura del documento

El documento se ha estructurado siguiendo el esquema clásico de descripción teórica de los desarrollos, seguido de la aplicación práctica de los mismos. En primer lugar se ha realizado un estudio del estado del arte en temas relacionados con la detección y seguimiento de objetos dinámicos. Se han estudiado desarrollos teóricos en aplicaciones de robótica móvil así como algunas soluciones empresariales a sistemas robóticos de seguridad.

Los siguientes tres puntos del documento contienen todo el desarrollo teórico realizado en este proyecto: la detección, la identificación y la fusión de información. El primer paso en todo proceso, la detección implica el análisis de los datos de los sensores en dos instantes consecutivos y la extracción de los objetos y sus características. Este paso se encuentra descrito en el apartado 3. El siguiente apartado describe los dos algoritmos desarrollados para la identificación de los objetos dinámicos; es decir, el emparejamiento entre los objetos observados en un instante y los observados en los instantes anteriores. Por último, se describen dos procedimientos distintos para la fusión de la información entre varios robots, el modo centralizado en el que la información fluye hacia y desde una base central y el modo descentralizado en el que los robots se comunican directamente entre sí.

Por último se incluyen las simulaciones realizadas en el entorno del proyecto NM-RS y los resultados obtenidos así como las conclusiones extraídas de todo el proyecto y las futuras líneas de investigación propuestas.

También se incluyen todas las referencias bibliográficas utilizadas para el desarrollo de los algoritmos y la redacción de esta memoria.

2. ESTADO DEL ARTE

La detección y seguimiento de objetos móviles se ha convertido en un campo de vital importancia para el correcto desarrollo de multitud de proyectos. Esta necesidad proviene de la influencia de este campo en aplicaciones tan diversas como supervisión de tráfico [7], navegación autónoma de robots y otros vehículos [8, 9], vigilancia autónoma de perímetros o instalaciones industriales [10] e incluso interacción entre robots y humanos [11].

Si se centra el foco de interés en las aplicaciones de robótica se puede observar, que a medida que progresaban los avances en robótica móvil, y en general en las aplicaciones que requerían una mayor autonomía del robot, el requisito de detección y seguimiento de objetos móviles se convertía más en una exigencia que en un añadido a la funcionalidad del robot.

Para los sistemas de seguridad estas nuevas funcionalidades se están convirtiendo rápidamente en críticas a la hora de diseñar sus plataformas. Si hasta ahora estos sistemas se conformaban con la detección de movimiento dentro de una zona de vigilancia, delegando en un operador humano la tarea de clasificar el nivel de riesgo; las nuevas aplicaciones obligan a que estos sistemas de seguridad sean capaces de discernir entre diferentes objetos móviles dentro de su rango de acción así como identificar la trayectoria de cada uno de ellos.

2.1. Reconocimiento de objetos dinámicos.

La clasificación de los diferentes desarrollos analizados para el reconocimiento de objetos dinámicos no es sencilla puesto que muchos de ellos difieren tanto en los métodos de procesado de la información del entorno, como en los algoritmos de reconocimiento de objetos empleados. Sin embargo, todos ellos pueden dividirse en dos grandes grupos en función del sensor utilizado para obtener la información del entorno: las aplicaciones basadas en cámaras de vídeo y visión artificial, y aquellas basadas en sensores láser con barrido horizontal. Existen otros desarrollos basados en otro tipo de sensores como sensores de ultrasonidos, pero que se centran en determinar si existe movimiento en la zona de vigilancia, no de reconocer el objeto móvil y extraer sus características principales [12].

El reconocimiento de objetos móviles en secuencias de video entraña generalmente un grado de dificultad muy grande y las herramientas disponibles actualmente están sujetas a restricciones que tienen que ver con el entorno en el que se realiza la detección, las características de los objetos móviles y su relevancia con respecto al ambiente. Los principales desarrollos para la detección en tiempo real de

objetos dinámicos se basan en diferencias de fotogramas sucesivos [13, 14], empleo de técnicas de contornos activos del objeto y redes neuronales para el análisis del movimiento [15], o detección de movimiento seguida de sucesivas segmentaciones en las regiones de interés [16, 17]. Además también se ha experimentado con la fusión de imágenes visibles e infrarrojas para mejorar las aplicaciones de reconocimiento de objetos móviles [18].

En cuanto a los desarrollos basados en láser existen diversos métodos con diferente número de sensores y diferentes técnicas de análisis de datos. Por ejemplo, Fod et.al. [19] utilizan una serie de sensores colocados en diferentes posiciones para seguir objetos móviles, en este caso personas, prediciendo las posiciones futuras con un modelo de movimiento. Sin embargo, el uso más común de estos sensores para la detección de objetos móviles es instaladas en alguna plataforma robótica, tanto estática como móvil. Chen et.al. [20] utilizan un sensor láser montado en un robot MORCS y tres mapas sucesivos en los que se comparan el estado de cada celda del mapa así como las 8 celdas colindantes para determinar los objetos dinámicos presentes en el entorno. Otro desarrollo en el que se utiliza un sensor sobre una plataforma móvil es el de Montmerlo et.al. [21] que utiliza un filtro de partículas condicional para detectar personas en movimiento.

Para decidir entre un desarrollo basada en cámara y una basada en láser es conveniente analizar las ventajas e inconvenientes de cada una de ellas, empezando por las primeras.

Aunque las aplicaciones basadas en cámaras han experimentado notables mejoras siguen presentando una serie de problemas que dificultan su implantación en todo tipo de entornos. Los principales problemas a los que se enfrentan este tipo de desarrollos son [22]:

- La pérdida de información por la proyección de un mundo tridimensional a una representación plana. Esto se acentúa en los desarrollos con láser puesto que en la mayoría la información que se obtiene del entorno proviene de un barrido con el láser en un solo plano horizontal.
- Ruido en las imágenes. Esto también ocurre en los láseres aunque en general son más robustos a la hora de resistir interferencias.
- Complejidad de los movimientos y los objetos. La mayor parte de los desarrollos en video se centran en determinados tipos de objetos basados en plantillas de reconocimiento que se aplican en las imágenes. Procesar todo tipo de objetos y movimientos resulta muy complejo con el nivel de detalle que proporcionan las cámaras.

- Cambios en la iluminación del entorno. Las cámaras son muy sensibles a la iluminación, necesitando unas condiciones precisas para su correcto funcionamiento y siendo vulnerables a cambios bruscos en ellas. Los láseres, sin embargo, no tienen este problema siendo su funcionamiento invariante ante los cambios de iluminación, funcionando incluso en ausencia de luz.
- Oclusiones parciales y totales. Es uno de los principales problemas en el seguimiento de objetos dinámicos, tanto en desarrollos basados en visión como en los basados en láser. No tiene una solución sencilla y todas pasan por predicciones más o menos acertadas del movimiento de los objetos implicados.

Los sistemas láser también presentan algunos problemas propios aparte de los ya discutidos para los sistemas con cámaras; que, aunque de menor importancia, también deben ser considerados:

- La sensibilidad del sensor varía dependiendo de algunos factores como el material del objetivo, la rugosidad, el color o la curvatura de la superficie
- No se obtiene información de color ni de forma, tan sólo se puede obtener una medida aproximada del tamaño del objeto, lo cual puede limitar los entornos de aplicación.

Debido a estas razones se ha optado por desarrollar esta plataforma basada en telemetría láser, que a pesar de presentar grandes ventajas con respecto a los desarrollos basados en cámara, también presenta algunos obstáculos para los que habrá que encontrar la mejor solución posible.

2.2. Reconocimientos de objetos dinámicos mediante telemetría Láser.

Existen numerosos desarrollos propuestos para seguimiento de objetos dinámicos y todos ellos se diferencian en la forma de tratar dos parámetros principalmente: el tipo de objeto que se pretende detectar y el modelado del movimiento. Sin embargo, en este trabajo se va a tratar de desarrollar una aplicación basada en láser que pueda detectar y seguir todo tipo de objetos sin importar forma y tamaño, tomando como objeto de interés cualquier móvil dentro del rango de detección [23]. Por lo tanto, se necesita hacer un pequeño estudio de las aplicaciones basadas en láser ya existentes que no discriminan el tipo de objeto móvil. Todas estas aplicaciones utilizan diferentes métodos de seguimiento y de representación del espacio.

Una distinción importante a tener en cuenta en los algoritmos de detección de objetos móviles es si el sensor se encuentra instalado sobre una plataforma estática o sobre un elemento móvil, como por ejemplo un robot.

La principal ventaja del uso de sensores estáticos es que la detección de movimiento es mucho más sencilla. Al mantenerse el fondo estático detectar los puntos del escáner que forman parte de los objetos móviles se consigue con una simple diferencia entre dos escáneres consecutivos. Sin embargo, en plataformas robóticas de vigilancia esto implica la instalación de un gran número de sensores en el caso de que la zona a vigilar sea muy extensa. Una posible solución es emplear robots móviles que vayan haciendo paradas para detectar posibles intrusos, aunque la solución ideal sería disponer de plataformas que pudiesen realizar la detección y la identificación de los objetos móviles en movimiento.

El principal inconveniente de la detección con robots móviles es la dificultad para distinguir los cambios que se producen en el escáner debido al propio movimiento del robot y los cambios producidos por los objetos dentro de la zona de vigilancia. Bobruk et.al [23] utilizan una compensación por movimiento en la que emparejan los puntos de los escáneres mediante un algoritmo ICP que devuelve una rotación y una translación. Estos valores de rotación y traslación se combinan con los datos de odometría y mientras los movimientos se limiten en velocidad (tanto de traslación y de rotación), la fusión proporciona una buena estimación del movimiento real de la plataforma.

Otra posibilidad usada por Chen [20] es utilizar técnicas de SLAM para crear mapas a medida que se mueve el robot y luego comparar las adiciones al mapa para extraer de ellas los objetos dinámicos. Para ello comparan el estado de cada celda en tres mapas de celdillas consecutivos, así como las ocho celdillas colindantes identificando de esta forma los objetos estáticos. Los objetos dinámicos se obtienen, por lo tanto, al comparar el mapa de objetos estáticos con el mapa obtenido del escáner actual.

2.2.1. Escáneres Láser.

Uno de los puntos de mayor importancia en este tipo de desarrollos es el tipo y rango del sensor utilizado para obtener la información del entorno. Los más utilizados son escáneres de proximidad basados en tecnología *LIDAR (Laser Imaging Detection and Ranging)* que permite determinar la distancia desde un emisor láser a un objeto o superficie usando un haz láser pulsado. Utiliza un sensor optoelectrónico que es capaz de proporcionar las indicaciones eléctricas de la distancia entre dos puntos de

referencia. La distancia se determina midiendo el tiempo de vuelo del haz de luz; es decir, el intervalo de tiempo entre la emisión del pulso y la detección de la señal reflejada. Estos sensores utilizan una zona del espectro de longitud de onda inferior a la del radar, por lo que es capaz de detectar pequeñas partículas.

El principio de funcionamiento de la mayor parte de los modelos comerciales se basa en un rayo láser reflejado en un espejo rotatorio, permitiendo así al sistema tomar varias medidas en un mismo plano. Usando el tiempo de vuelo y el ángulo del espejo para cada medida se puede determinar la distancia desde el emisor a todos los objetos del entorno [24]. El esquema de un sensor LIDAR se puede observar en la Figura 1.

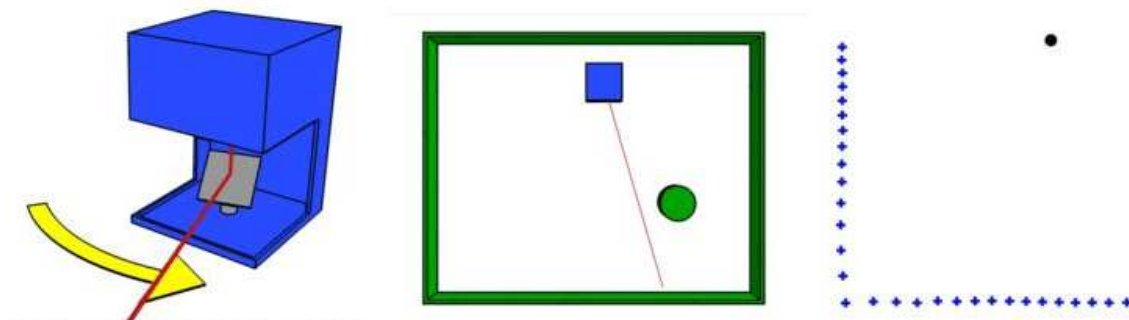


Figura 1: Esquema de un sensor LIDAR

Dependiendo del sensor escogido se dispone de una gran variedad de rangos, resoluciones y ángulos de visión; sin embargo todos ellos devuelven la distancia y el ángulo para cada punto. Estos sistemas tienen algunas ventajas importantes frente a otros sistemas como los ultrasonidos o los sistemas de visión. La principal ventaja es la alta precisión y rango de que disponen; pueden medir desde los 0 a los 80 metros con una precisión que llega a los milímetros. Además, no pierden sensibilidad al acercarse mucho el objeto al punto de emisión del láser, inconveniente que si está presente en las cámaras, que necesitan una distancia mínima para enfocar. Otra ventaja reseñable es que no están restringidos por las condiciones lumínicas; es decir, se pueden utilizar tanto de día como de noche.

El mayor inconveniente que presentan los sistemas de telemetría láser es el alto precio que tienen estos sensores que los pueden hacer inviables para las aplicaciones más simples. Otro factor influyente a la hora de decidirse por ese tipo de sistemas es el peso. Se trata de sensores que pueden pesar entre 3 y 5 kg y que los pueden hacer desaconsejables para aplicaciones con robots pequeños. La utilización de láseres puede ser considerada peligrosa en función del tipo de rayo utilizado, pero cada vez más este tipo de sensores utilizan láseres de tipo 1 que no son peligrosos y pueden ser usados en ambientes con interacción humana.

Otro inconveniente a considerar es el tiempo de realización de una medida completa en los sistemas LIDAR. Si además se incorpora un mecanismo para hacer medidas en varios planos colineales de forma que se realice un escáner 3D, estos tiempos pueden suponer un problema para usar este tipo de escáneres en aplicaciones en tiempo real. Una posible solución a este problema sería la utilización de dispositivos FLASH LIDAR que no utilizan un pulsos de luz concentrada en un solo punto, para luego escanear mecánicamente el entorno. Estos sistemas inundan la escena con luz de láser difusa y usan un conjunto de planos focales para obtener una imagen 3D cada vez que se dispara el láser. La principal ventaja de estos sistemas es la velocidad de la obtención de datos. En lugar de obtener punto por punto la distancia a todos los objetos de entorno, con la consecuente distorsión debida al movimiento del escáner, se obtiene la imagen completa de una sola vez. Estos sistemas también son conocidos como LADAR (*Laser Detection and Ranging*).

Todos los sensores láser producen medidas de distancia, que deben ser analizadas para extraer de ellas la mayor cantidad de información posible. Existen diferentes técnicas de análisis, que se describen a continuación.

2.2.2. Técnicas de análisis aplicadas

De las diferentes técnicas existentes en el análisis de los datos de los sensores láser, una de las más utilizadas para transformar los datos del láser en información útil sobre el entorno son las matrices de ocupación probabilística. Se utilizan este tipo de matrices para ser capaz de identificar y seguir objetos en el entorno.

Matrices de ocupación probabilística

Las matrices de ocupación probabilística representan un mapa del entorno como una serie de variables uniformemente espaciadas en la que cada una representa la presencia de un obstáculo en ese lugar del entorno del robot. Los mapas probabilísticos más comunes son representaciones en 2D de un plano del entorno 3D del robot. La idea principal detrás de esta representación es convertir el entorno en un mapa de celdillas en el que en cada celda se introduce el un valor que indica la probabilidad de que esa celda esté ocupada. Estos mapas se construyen alrededor de la posición del láser ya que se conocen las distancias desde el sensor hasta los obstáculos [23]. Un ejemplo de mapa de celdas se puede ver en la Figura 2, donde el robot se sitúa en el centro de la imagen.

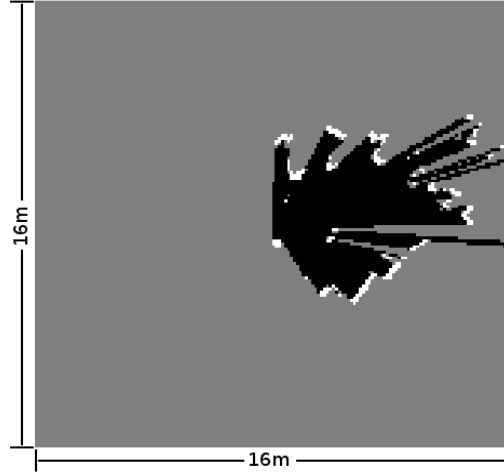


Figura 2: Ejemplo de mapa de celdas

La construcción de estos mapas de celdas varía según el tipo de sensor utilizado, pero el proceso básico es siempre el mismo [20]. En primer lugar se calcula la posición de los objetos en las coordenadas globales del sistema:

$$x_g = x_r + \rho \cdot \cos(\theta + \alpha)$$

$$y_g = y_r + \rho \cdot \sin(\theta + \alpha)$$

donde (x_g, y_g) son las coordenadas globales del obstáculo; (ρ, α) son la distancia al objeto y el ángulo del láser correspondiente; y (x_r, y_r, θ) la posición y orientación del robot. Una vez conocida la posición global del obstáculo se debe proyectar esta posición en el mapa de celdillas, siendo w el ancho de cada celda.

$$x_{gm} = \text{int}(x_g/w) \cdot w + \text{int}(w/2)$$

$$y_{gm} = \text{int}(y_g/w) \cdot w + \text{int}(w/2)$$

El principal método para la detección de movimiento es la diferencia directa entre dos mapas de ocupación probabilística [23]. Cuando se genera un mapa de diferencias hay que tener en cuenta que sólo las celdas ocupadas en el mapa actual son relevantes a la hora de detectar movimiento delante del robot. Cualquier otro cambio puede deberse o bien a objetos antes visibles pero que ahora han quedado ocluidos o a objetos previamente detectados (las celdas ya estarían marcadas como ocupadas). El cálculo del mapa de diferencias puede resumirse mediante la siguiente fórmula, donde G es el mapa actual y G' es el mapa anterior:

$$D_{ij} = \begin{cases} 1, & \text{si } G_{ij} = 1 \text{ y } G'_{ij} = 0 \\ 0, & \text{en caso contrario} \end{cases}$$

Una vez se conocen los puntos en los que existe diferencia entre los dos mapas se pueden agrupar para efectuar un correcto seguimiento de los objetos

móviles detectados por el escáner. De la diferencia de estos dos mapas se puede obtener una visión del número y dirección de los objetos presentes en la imagen.

Sin embargo, extrapolar esta técnica a un escáner 3D no es un proceso automático. Existen varias dificultades que impiden que se pueda aplicar este método directamente. Cada medida realizada por un escáner 3D se compone de una serie de escáneres en 2D que se realizan haciendo girar el escáner sobre su eje horizontal. Sólo uno de esos escáneres es paralelo al suelo, mientras que el resto conforman un ángulo que aumenta a medida que el escáner se inclina más con respecto al plano central. Si un escáner 2D se transforma en una matriz probabilística, un escáner 3D debería transformarse en tantas matrices como escáneres 2D se hayan realizado en toda la medida. Sin embargo, la transformación de los datos no es inmediata puesto que se deben incluir los ángulos de *pitch* y *yaw* en el cálculo de los puntos de colisión. Otra razón por la que este método no es adecuado es la cantidad de datos que habría que manejar; el número normal de elementos de la matriz aumentaría exponencialmente puesto que habría que utilizar una matriz para cada escáner 2D realizado.

Segmentación de los escáneres

Katz et.al. [25] abordan el problema desde otra perspectiva. En lugar de convertir cada escáner en un mapa del entorno se inclinan por segmentar cada medida y obtener entidades simples que ayuden a simplificar la representación. La segmentación se realiza mediante la comparación de medidas consecutivas que llegan en el orden de ángulo de barrido creciente. Además se plantea esta segmentación en un espacio euclídeo donde la distancia umbral se usa como criterio para agrupar los núcleos.

Wang [26] utiliza una implementación conjunta de SLAM y DATMO (*Detection and Tracking of Moving Objects*) en el que el primer paso de la detección, una vez se dispone de una buena estimación de la posición y un mapa del entorno generado por el SLAM, es segmentar cada escáner usando un criterio de distancia. Este criterio obliga a que todos los puntos que estén a menos de un metro de distancia formen parte del mismo segmento. Los objetos móviles se detectan buscando inconsistencias entre el mapa y el escáner, aunque esta hipótesis no es del todo correcta pues existen excepciones provocadas generalmente por las oclusiones parciales o totales. Por lo tanto el algoritmo de detección de objetos móviles consta de dos partes; en la primera se detectan los puntos del escáner que se consideran móviles y en la segunda parte se combina esta información con la información generada por la segmentación para decidir que grupos forman un objeto móvil. Para la identificación y el seguimiento de

los objetos móviles utilizan un algoritmo IMM (*Interacting Multiple Model*) suponiendo movimientos de velocidad o aceleración constante.

2.3. Seguimiento de objetos

2.3.1. Técnicas utilizadas en sistemas de un solo robot

El principal elemento diferenciador en los algoritmos de detección y seguimiento de objetos móviles es el método elegido para identificar y seguir a los objetos. Como ejemplo se van a describir dos de ellos que usan técnicas diferentes.

Uno de los métodos más empleados consiste en la segmentación del mapa de celdillas en los diferentes objetos seguida del uso de alguna variante del filtro de Kalman para identificar a los objetos y eliminar el ruido. Bobruk y Austin [23] utilizan un algoritmo de “segmentación de manchas” (*blob segmentation*) para separar cada uno de los objetos encontrados en los que se identifica cada mancha por una serie de parámetros como su posición o su tamaño. Si bien el centro de masas que se identifica no es el centro de masa del objeto debido a que con los sensores láser sólo se obtiene información de profundidad, ésta es suficiente para ser capaz de seguir la cara del objeto como si fuera el objeto en su totalidad. Para manejar los diferentes objetos que van apareciendo se crean “hipótesis” de movimiento que contienen información de la posición y la velocidad de un objeto; siempre que la misma mancha se haya identificado correctamente en al menos dos escáneres consecutivos. Para evitar el aumento del número de hipótesis con las que se trabaja, que después aumentarían mucho el grado del filtro de Kalman con el que habría que trabajar, cada nueva “mancha” que se detecta pasa a ser una nueva hipótesis o a formar parte de una hipótesis existente. La decisión se toma asignando a cada mancha un peso específico y tomar como hipótesis nuevas a las manchas con un peso mayor a un límite establecido de forma experimental.

Las hipótesis almacenan y se actualizan con la posición y velocidad estimada de los objetos; así como un peso específico que indica la posibilidad de que se esté siguiendo un objeto móvil. Este peso se establece inicialmente a un nivel bajo y se va aumentando si la evolución del objeto móvil se corresponde con el modelo de objeto móvil a velocidad constante que se usa en la predicción de velocidad y posición. El uso de hipótesis tiene otras ventajas que ayudan a dar cobertura a determinados casos. Por ejemplo, en el caso de que se distingan dos “manchas” diferentes cercanas a una misma hipótesis, ésta se divide permitiendo seguir los dos movimientos de forma independiente. Así mismo, si una hipótesis se queda sin ninguna mancha, ésta no se elimina; su peso específico va disminuyéndose paulatinamente, permitiendo

manejar así las oclusiones y evitando marcar como nuevo un objeto que en realidad sólo estaba oculto. Para manejar y seguir todas estas hipótesis se utiliza el filtro de Kalman, eligiendo los parámetros de tal forma que se recompensan los objetos moviéndose a velocidad constante y que se filtre lo más posible el ruido. Este proceso funciona tanto con el sensor estacionario como montado sobre una plataforma móvil, siempre que este movimiento no sea muy brusco o rápido.

Otra posibilidad para realizar el seguimiento de objetos dinámicos es la utilización de un algoritmo ICP (*Iterative Closest Point*) [27] para conseguir correspondencias espacio- temporales entre los diferentes escáneres realizados con el láser; seguida de una representación probabilística basada en núcleos, incluyendo integración de múltiples escáneres y comprobación para oclusiones. El principal problema al que se enfrentan este tipo de algoritmo es la asociación de puntos pertenecientes a los objetos estáticos para poder medir su desplazamiento de posición en los escáneres. Una vez se conoce esta información, cualquier otro cambio que se produzca en los escáneres se deberá sin duda a objetos dinámicos dentro del rango de acción del sensor. El proceso seguido necesita primero de una segmentación y un registro de los escáneres, seguida de una representación probabilística de los núcleos presentes en cada uno de los escáneres.

La segmentación se usa para dividir los escáneres en entidades más pequeñas que simplifican la representación. El proceso de alineamiento o registro de los núcleos creados en la fase anterior con un punto de referencia se puede realizar de diferentes formas siendo la más utilizada los algoritmos ICP. Estos algoritmos se utilizan para minimizar las diferencias entre dos nubes de puntos y se basa en ir revisando iterativamente la transformación (traslación y rotación) necesaria para minimizar puntos entre dos escáneres cualesquiera. Los pasos a seguir para utilizar este tipo de algoritmos son [27]:

- Asociar los puntos más cercanos entre sí.
- Estimar los parámetros de la transformación usando una función de coste cuadrático medio.
- Transformar los puntos usando los parámetros estimados.
- Iterar volviendo a asociar los puntos.

Esta correspondencia se basa en asumir que la mayor parte de la información recabada por el láser proviene de elementos estáticos del entorno. Las correspondencias obtenidas se usan para clasificar los núcleos identificados entre estáticos y dinámicos, así como su desplazamiento. Esto se realiza mediante una sincronización probabilística usando la media y la covarianza gaussiana.

2.3.2. Técnicas utilizadas en sistemas multirobot

En el caso de sistemas multirobot, existe una capa superior de control en la que se produce la fusión de la información generada por cada robot, permitiendo una mejor identificación de cada objeto móvil. Compartir la información entre los robots puede aumentar de forma efectiva la visibilidad instantánea del entorno y permite la creación de mejores modelos y una mejor respuesta ante cualquier cambio.

Stroupe y Martin [28] presentan un método para representar, comunicar y fusionar distintas medidas de un objeto obtenidas por diversos robots. Utilizan una distribución gaussiana de dos dimensiones como representación para cada observación simple de un objeto. El centro de la distribución es la posición estimada del objeto y las desviaciones estándar alrededor del eje mayor y menor son las estimaciones de la incertidumbre en cada eje. Para fusionar dos medidas utilizan un procedimiento estadístico basado en la regla de Bayes y el filtro de Kalman, que permite la fusión de varias medidas de forma simultánea en cualquier orden. Este método se basa en algunas hipótesis que pueden desembocar en problemas a la hora de aplicar el algoritmo de forma global. En primer lugar asumen que los sensores son independientes y con una distribución normal y que los ejes de coordenadas de los robots son coincidentes. Aunque estas dos hipótesis pueden considerarse ciertas en el entorno en el que se desarrollan las pruebas, también suponen que los robots están perfectamente localizados. Esto puede producir problemas en la estimación de la posición de los objetos. Sin embargo, modelando estos errores como gaussianos, se pueden tener en cuenta de forma similar a la forma en la que se fusionan las medidas de los objetos. Por último, también se asume que las medidas son simultáneas, debido a que las medidas obtenidas por los sensores suelen ser muy rápidas y a que se supone que los objetos no se mueven a velocidades muy altas. El mayor problema de este método es, sin embargo, que supone que los objetos son únicos para evitar el problema de la asociación de las medidas con cada uno de los objetos. Por lo tanto este sistema no sería válido para un entorno de múltiples objetos dinámicos en el que no se conoce ni el número ni la morfología de ninguno de ellos.

Otro enfoque a este problema es el utilizado por Mazo et.al. [29] que plantean un algoritmo jerárquico para localizar y seguir un objeto dinámico. En este caso estudian el caso de un equipo de dos robots que deben seguir a un solo objeto móvil. Estos robots tienen fuertes restricciones en los sensores usados y en el rango de estos. Las limitaciones presentes obligan a desarrollar un esquema cooperativo en el que un algoritmo de alto nivel planifica una formación para que los robots puedan seguir el objetivo. En este caso los robots intercambian información de los sensores para poder

estimar la posición del móvil mediante triangulación. Aunque el algoritmo resuelve el problema de forma eficiente, tampoco es aplicable al caso que se plantea ya que, de nuevo, sólo aborda el seguimiento de un objeto móvil y maneja muy poca información sensorial.

Por último, hay que reseñar el trabajo realizado por Göhring y Burkhard [30]. El principal problema que resuelven es el de la doble inclusión de errores al transformar el sistema de coordenadas basadas en el robot que usan los sistemas de un solo elemento a un sistema de coordenadas global. Los dos errores que se suman son el error de localización del objeto por parte del robot que se ve amplificado por el error de localización del robot en el sistema de coordenadas global. Proponen utilizar la relación entre los diferentes objetos detectados en una imagen como base para determinar la posición de uno de ellos en el sistema de coordenadas global. La relación de los objetos dinámicos con respecto a los estáticos facilita que los robots puedan determinar la posición de un objeto dinámico en el sistema de coordenadas global a pesar de no conocer con exactitud su propia posición y luego utilizar esa información para mejorar el cálculo de su posición en el entorno.

2.4. Información utilizada para la identificación de objetos dinámicos.

2.4.1. Datos del láser. Matrices de distancia.

Los datos que devuelve el láser se obtienen en forma de matriz en la que cada elemento indica la distancia hasta el objeto que el rayo correspondiente ha intersecado.

$$D = \begin{bmatrix} D_{11} & \dots & \dots & D_{m1} \\ \vdots & D_{ij} & \ddots & \vdots \\ \vdots & \ddots & \ddots & \vdots \\ D_{n1} & \dots & \dots & D_{nm} \end{bmatrix}$$

Así mismo se conocen los ángulos que tiene cada uno de los rayos que parten del sensor; por lo que, conociendo la posición del sensor, se puede calcular la posición del punto de colisión del rayo con el objeto.

La representación por colores de una matriz de distancia se observa en la Figura 3 y proporciona una representación del “mundo” situado enfrente del robot. En este caso se observa la línea del horizonte (curvada debido a que el rango de medida horizontal es mayor que el vertical) y tres formas situadas en el centro que se

corresponden con tres objetos dentro del rango de medida del sensor. El movimiento de estos objetos se determinará con los algoritmos que descritos más adelante.

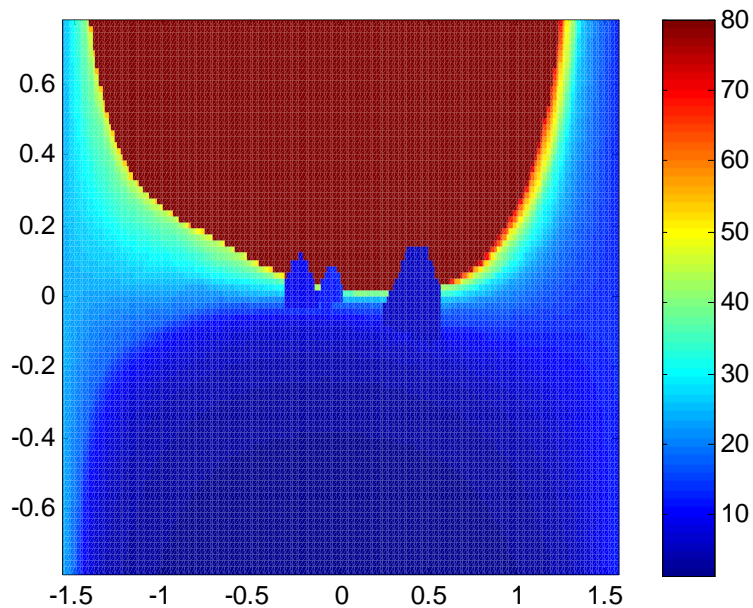


Figura 3: Representación por colores de una matriz de distancias

2.4.2. Datos de la cámara y la cámara infrarroja. Extracción del color y la temperatura.

El proceso de extracción de los datos de la cámara y la cámara infrarroja es similar y en ambos casos necesita de una calibración de ambos dispositivos. Las medidas de distancia que produce el láser llevan asociadas automáticamente unos ángulos que permiten localizar inequívocamente el punto del espacio en el que el rayo ha impactado. Sin embargo, esta exactitud no es igual para las cámaras, puesto que sólo proporcionan una imagen; es decir, una matriz en la que cada punto (o pixel) indica el color (o colores en caso de que proporcione los tres canales RGB) o la temperatura (en el caso de la cámara termográfica).

Por lo tanto es necesario conocer el campo de visión de ambas cámaras y corresponder cada punto de la cámara con un punto del láser, teniendo en cuenta las resoluciones de ambas matrices. Actualmente ya existen cámaras unidireccionales que tienen un campo de visión horizontal de 360° por lo que no existirán puntos en los que la información del láser no se pueda complementar con las de las cámaras.

3. ALGORITMOS DE DETECCIÓN DE OBJETOS MÓVILES

3.1. Detección de movimiento.

La detección de movimiento se hace por diferenciación directa de las matrices de distancia. Se obtiene una matriz de diferencias a la que se le pasa un filtro para detectar los puntos que han cambiado por encima de un determinado umbral. En la Figura 4 se observa un ejemplo de la diferenciación de matrices para la extracción de los objetos dinámicos.

$$Dif = \begin{cases} 1, & \text{si } abs(D_{if} - D'_{ij}) > Dif_{lim} \\ 0, & \text{si } abs(D_{if} - D'_{ij}) < Dif_{lim} \end{cases}$$

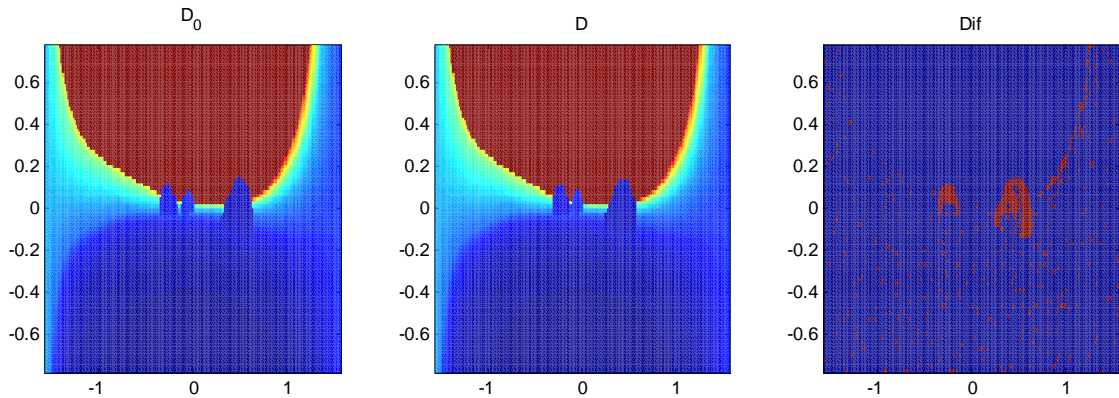


Figura 4: Ejemplo de diferencia de matrices de distancias

Mediante este sencillo filtro se consigue obtener una matriz donde las zonas no nulas son las indicadoras de movimiento. Sin embargo, no todos esos puntos se corresponden con objetos móviles. Para discernir entre objetos móviles y discrepancias producidas por errores de medida se deben agrupar los puntos con movimiento entre sí, formando objetos.

El primer paso es agrupar todos los puntos que estén en contacto entre sí, en este caso se ha utilizado una conectividad a 8, y asignarles una etiqueta común, identificándolos como un mismo objeto. En este paso también se descartan objetos que estén compuestos únicamente por un punto. Estos puntos pueden ser producidos por errores de medida, por objetos muy pequeños o por objetos muy alejados del robot. En los dos últimos casos se pueden eliminar los puntos sin incurrir a grandes errores debido a que estos objetos no son relevantes para la planificación de

desplazamiento del robot. Además se ha comprobado que la mayor parte de estos puntos son producidos por errores entre las medidas del láser. Finalizado este proceso se obtiene una matriz donde quedan resaltados los puntos que se han considerado como objetos móviles, como se observa en la Figura 5.

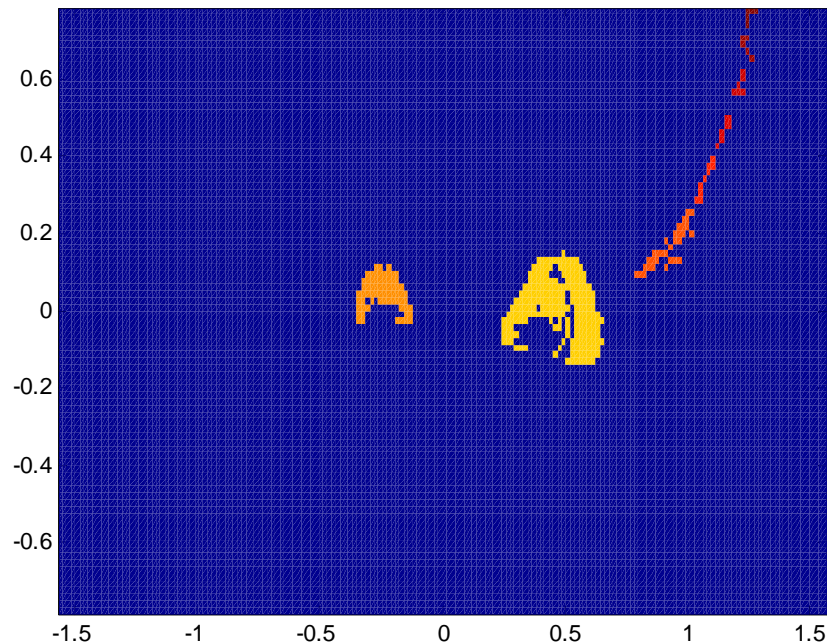


Figura 5: Ejemplo de matriz tras el proceso de filtrado

Una vez obtenida esta matriz se deben extraer algunas de las características principales de los objetos identificados:

- Posición del centro del objeto: Se obtiene haciendo la media de las coordenadas de todos los puntos de colisión.
- Coordenadas de la celda central: Coordenadas dentro de la matriz de la celda que contiene al punto identificado como posición central del objeto.
- Tamaño del objeto: Se obtiene de la diferencia entre el máximo y mínimo valor para cada una de las coordenadas. Se calculan tres valores aunque el valor de profundidad no es significativo puesto que con el láser no es posible percibir la tercera dimensión del objeto correctamente.

Para la realización de este proceso se deben convertir cada uno de los puntos del objeto a la posición real. Sin embargo, debido al proceso seguido para identificar todos los puntos cuyo valor ha cambiado, se han incluido también puntos que no pertenecen actualmente al objeto móvil. Un ejemplo claro serían los puntos que en la captura anterior formaban parte del objeto pero en la captura actual han dejado de

formar parte de él. Estos puntos se han incluido dentro del conjunto de puntos móviles, pero si se tienen en cuenta para el cálculo de la posición y el tamaño pueden inducir a error. Por lo tanto deben ser eliminados antes de realizar este cálculo. Para ello se comparan los valores de cada uno de los puntos de cada objeto con los valores de la matriz de referencia y se descartan los puntos en los que la diferencia sea menor que un umbral previamente establecido debido a que esos puntos seguramente pertenecen al fondo.

Esta matriz de referencia se va modificando dinámicamente con cada lectura del láser recibida tal y como se explica en la siguiente sección.

3.2. Creación y actualización de la matriz de referencia.

La matriz de referencia, o matriz de fondo, se utiliza para mantener una imagen del fondo estático sobre el que se mueven los objetos dinámicos. Cada uno de los puntos de la matriz indica la distancia al elemento del entorno estático que interseca con ese rayo.

El método ideal para la creación de esta matriz de referencia sería el cálculo de la media entre tres imágenes consecutivas tomadas sin ningún objeto móvil dentro de la imagen. Una vez calculada esta matriz media se mantendría como la matriz de fondo hasta que el robot cambiase de posición, momento en el que habría que repetir la operación para obtener una nueva matriz. Este esquema de creación y actualización presenta un problema muy grave; y es que en el momento de la realización de las medidas haya presente delante del robot algún elemento que más adelante vaya a cambiar su posición, por ejemplo, otro robot que en ese momento se encuentre parado. Al tomarse solamente medidas en un instante, cualquier cambio posterior no quedaría reflejado en la matriz de referencia. A medida que estos cambios sean más pronunciados se introducirían más errores en el cálculo de las dimensiones reales de los objetos.

Para evitar estos problemas se ha utilizado un método de actualización constante de la matriz de forma que la imagen de referencia se va modificando a medida que el entorno del robot cambia. Existen varias formas de realizar este proceso de actualización, pero en este caso se ha utilizado un método basado en la media y la varianza de los últimos tres valores para cada punto. Para ello se mantiene una matriz de medias y varianzas de referencia. La matriz de medias de referencia (D_{ref}) es la que se utiliza como imagen de referencia para el fondo mientras que la matriz de varianzas de referencia (V_{ref}) se utiliza para observar la tendencia en los

cambios y saber cuándo hay que sustituir el valor actual de D_{ref} por la nueva media calculada.

El proceso a seguir es el siguiente:

- Se calcula la media y la varianza de los últimos tres valores para cada punto.

$$\overline{D_{ij}} = \sum_{k=1}^3 D_{ij}^k \quad V_{ij} = \sum_{k=1}^3 (D_{ij}^k - \overline{D_{ij}})^2$$

- En el caso de que la varianza sea menor que el valor de la varianza guardado en V_{ref} , se actualizan los valores de referencia, así se eliminan, o por lo menos se reducen, los errores en la medida.

$$Si V_{ij} < V_{ij}^{ref} \rightarrow D_{ij}^{ref} = \overline{D_{ij}} \quad y \quad V_{ij}^{ref} = V_{ij}$$

- En el caso de que la media sea mayor que la media de referencia, se actualizarán los valores sólo si la varianza medida dividida entre cuatro es menor que la varianza de referencia.

$$Si (V_{ij} < 4 \cdot V_{ij}^{ref} \quad y \quad \overline{D_{ij}} > D_{ij}^{ref}) \rightarrow D_{ij}^{ref} = \overline{D_{ij}} \quad y \quad V_{ij}^{ref} = V_{ij}$$

- En caso contrario se mantienen los valores iniciales de la media y la varianza de referencia.

Todo este proceso se repite para cada punto de la matriz con cada medida proveniente del láser. De esta forma se mantiene una matriz de referencia actualizada en la que se mantienen con mayor preferencia los valores más grandes, ya que son los que corresponderán al fondo en su mayor parte. Sin embargo, también se podrían sustituir valores más grandes por valores más pequeños (objetos más cercanos) si estos permanecen mucho tiempo estáticos. Esto tiene especial interés en el caso de que determinados móviles se introduzcan en el rango de acción del sensor y seguidamente se queden inmóviles el suficiente tiempo como para considerarlos parte del entorno.

3.3. Matrices de colisión.

La extracción de todas las características reales de los objetos necesita el cálculo de la matriz de colisión. Esta matriz contiene cada una de las medidas del láser como un punto en el espacio. Conociendo los ángulos de inclinación (alfa y sigma) del rayo del láser para cada punto de la matriz, el ángulo yaw del sistema EOS, y conociendo la distancia que devolvió el láser se puede calcular la posición relativa de ese punto con respecto al punto de emisión del láser. Como también se conoce la

posición relativa del escáner láser respecto a la posición del robot en el espacio y la del robot con respecto al sistema de coordenadas global utilizado, es cuestión de realizar una serie de transformaciones para calcular la posición del punto de colisión del láser en el sistema de coordenadas global elegido.

Teniendo en cuenta que la matriz D representa una imagen en la que el valor de pixel describe la distancia al punto de impacto de cada uno de los rayos del láser, se tiene que la fila superior de la matriz corresponde al ángulo de barrido vertical más negativo y la fila inferior de la matriz corresponde al más positivo. Lo mismo se puede decir de la columna izquierda de la matriz que representaría el ángulo de barrido horizontal máximo mientras que la columna situada más a la derecha se representa el ángulo de barrido horizontal mínimo. De esta forma la matriz queda como:

$$D = \begin{bmatrix} (\alpha_{max}, \sigma_{min}) & \dots & (\alpha_{min}, \sigma_{min}) \\ \vdots & & \vdots \\ (\alpha_{max}, \sigma_{max}) & \dots & (\alpha_{min}, \sigma_{max}) \end{bmatrix}$$

Luego, para cada elemento (i,j) de la matriz, los ángulos asociados a cada medida de distancia son:

$$\alpha = \alpha_{max} - (j - 1) \cdot RH$$

$$\sigma = \sigma_{min} + (i - 1) \cdot RV$$

Donde,

- α y σ son los ángulos del láser según las referencias establecidas.
- α_{max} es el ángulo de apertura horizontal máximo correspondiente a la mitad del ángulo total de barrido horizontal.
- σ_{min} es el ángulo de apertura vertical mínimo correspondiente a la mitad del ángulo total de barrido vertical.
- RH y RV son las resoluciones angulares horizontal y vertical.

A continuación, se realiza la transformación de coordenadas esféricas a cartesianas según el sistema de referencia 3, centrado en la óptica del láser, según las siguientes expresiones:

$$X_3 = D_{ij} \cos(-\sigma_{ij}) \cos(\alpha_{ij})$$

$$Y_3 = D_{ij} \cos(-\sigma_{ij}) \sin(\alpha_{ij})$$

$$Z_3 = D_{ij} \sin(-\sigma_{ij})$$

Posteriormente, se debe pasar desde el sistema de referencia 3 o sistema del láser hasta el sistema fijo para representar cada punto de impacto respecto al sistema de coordenadas fijo. Los sistemas de referencia utilizados se encuentran definidos en la Figura 6. Para ello, se cuenta con la información de los ángulos de Euler del sistema

móvil (2) respecto al fijo (1) (actitud del robot) y se utiliza el cálculo matricial basado en transformaciones homogéneas. Para este cálculo se deben definir los ángulos de Euler utilizados:

- *Roll robot* (φ_r), ángulo que representa el giro del robot respecto al eje X_1 fijo.
- *Pitch robot* (θ_r), ángulo que representa el giro del robot respecto al eje Y_1 fijo.
- *Yaw robot* (ψ_r), ángulo que representa el giro del robot respecto al eje Z_1 fijo.
- *Roll láser* (φ_l), ángulo que representa el giro del escáner láser respecto al eje X_2 fijo.
- *Pitch láser* (θ_l), ángulo que representa el giro del escáner láser respecto al eje Y_2 fijo.
- *Yaw láser* (ψ_l), ángulo que representa el giro del escáner láser respecto al eje Z_2 fijo.

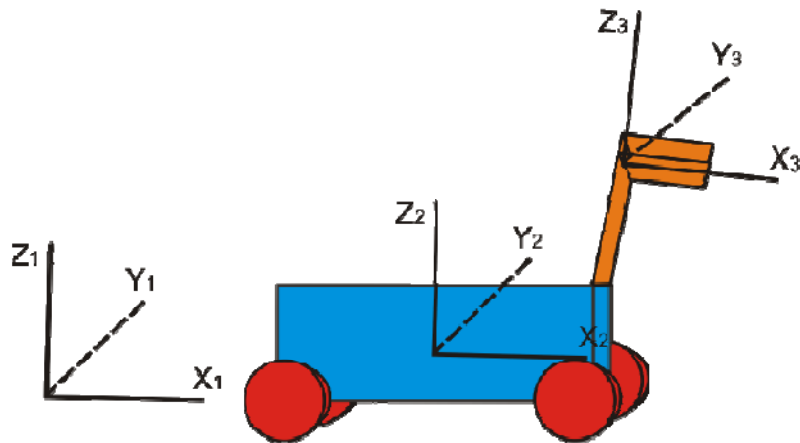


Figura 6: Sistemas de referencia empleados en el robot

Una vez definidos los ángulos de Euler sólo hay que especificar las operaciones necesarias para pasar la posición de un punto de colisión del sistema de coordenadas 3 al sistema de coordenadas 1. La conversión del sistema 3 al 2 se realiza mediante una translación y una rotación, puesto que la posición del mapa se encuentra desplazada con respecto al punto de robot que se toma como referencia para medir su posición; además es necesaria una rotación debido a que el escáner láser normalmente se encuentra montado sobre una plataforma que puede girar. El paso del sistema 2 al sistema 1 necesita de igual modo una translación y una rotación puesto que el robot se encuentra desplazado y girado con respecto al sistema de coordenadas global 1. Las operaciones necesarias se incluyen en la siguiente fórmula donde (X_p^1, Y_p^1, Z_p^1) es la posición del punto de colisión con respecto al sistema de coordenadas global 1; (X_p^3, Y_p^3, Z_p^3) la posición del punto de colisión con respecto al

sistema de coordenadas 3; (X_2, Y_2, Z_2) la posición del robot con respecto al sistema de coordenadas 1; y (X_3, Y_3, Z_3) la posición del escáner láser con respecto al sistema de coordenadas 2.

$$\begin{pmatrix} X_p^1 \\ Y_p^1 \\ Z_p^1 \\ 1 \end{pmatrix} = T_2^1 \cdot R_2^1 \cdot T_3^2 \cdot R_3^2 \cdot \begin{pmatrix} X_p^3 \\ Y_p^3 \\ Z_p^3 \\ 1 \end{pmatrix} =$$

$$\begin{pmatrix} 1 & 0 & 0 & X_2 \\ 0 & 1 & 0 & Y_2 \\ 0 & 0 & 1 & Z_2 \\ 0 & 0 & 0 & 1 \end{pmatrix} \cdot R(\theta_r, \psi_r, \varphi_r) \cdot \begin{pmatrix} 1 & 0 & 0 & X_3 \\ 0 & 1 & 0 & Y_3 \\ 0 & 0 & 1 & Z_3 \\ 0 & 0 & 0 & 1 \end{pmatrix} \cdot R(\theta_l, \psi_l, \varphi_l) \cdot \begin{pmatrix} X_p \\ Y_p \\ Z_p \\ 1 \end{pmatrix}$$

Las matrices de rotación representadas en la fórmula con $R(\theta, \psi, \varphi)$, se pueden calcular como:

$$R(\theta, \psi, \varphi) = \begin{pmatrix} C(\psi)C(\theta) & C(\psi)S(\theta)S(\psi) - S(\psi)C(\varphi) & C(\psi)S(\theta)C(\psi) + S(\psi)S(\varphi) & 0 \\ S(\psi)C(\theta) & C(\psi)S(\theta)S(\psi) + C(\psi)C(\varphi) & S(\psi)S(\theta)C(\psi) - C(\psi)S(\varphi) & 0 \\ -S(\theta) & C(\theta)S(\varphi) & C(\theta)C(\psi) & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

Una vez aplicada esta transformación matricial, se obtiene cada punto de impacto del láser respecto del sistema fijo y ya solo queda asignarlo a una determinada celda del mapa. Las celdas se han numerado siguiendo los criterios de numeración de los elementos de una matriz. Como ejemplo consideraremos que el centro del mapa se encuentra en la casilla (3,3). En este caso, toda variación hacia la parte positiva del eje X irá aumentando el número de la columna, y toda variación en el sentido del eje Y positivo reducirá el número de fila de la matriz, tal y como describen las siguientes expresiones:

$$j = \begin{cases} X + \text{int}\left(\frac{\Delta x}{\text{Escala}}\right) & \forall \Delta x > 0 \\ X + \text{int}\left(\frac{\Delta x}{\text{Escala}}\right) & \forall \Delta x < 0 \end{cases}$$

$$i = \begin{cases} Y - \text{int}\left(\frac{\Delta y}{\text{Escala}}\right) & \forall \Delta y > 0 \\ Y - \text{int}\left(\frac{\Delta y}{\text{Escala}} - 1\right) & \forall \Delta y < 0 \end{cases}$$

Donde,

- Δx e Δy son las distancias del punto de colisión al centro del mapa en la dirección de los ejes de coordenadas globales.
- *Escala* es la dimensión de la celda en metros.
- i y j son los números de la fila y la columna a la que corresponde el punto de colisión.

3.4. Extracción de características reales.

De las medidas del láser se extraen dos datos muy importantes para la correcta identificación de los objetos dinámicos, la posición y el tamaño.

El cálculo de la posición de un objeto móvil se realiza por media directa del número de celdas en cada una de las dos dimensiones de la matriz de distancia. El proceso que se sigue para la obtención de la posición en el plano a partir de todos los puntos de una matriz de distancia identificados como parte de un objeto es el siguiente:

- Se comprueba si el valor de la distancia para el punto (i,j) de la matriz (perteneciendo ese punto a los que se han identificado como parte del objeto) es similar al valor de la distancia en la matriz de referencia.

$$\text{Si } D(i,j) \approx D_{ref}(i,j) \rightarrow (i,j) \text{ se elimina de la lista}$$

- Una vez se conocen todos los puntos (i,j) de la matriz que sin duda pertenecen al objeto, se calcula la celda central del objeto:

$$i_{central} = \frac{\sum i}{n}, \quad j_{central} = \frac{\sum j}{n} \quad (\text{donde } n \text{ es el número de puntos})$$

- Con los valores de la celda central y la medida de distancia en ese punto se calcula la posición real del centro del objeto y su posición en el plano será la proyección de ese punto.

Para la determinación del tamaño del objeto se sigue un proceso similar al anterior. Una vez se determina de forma efectiva los puntos que pertenecen al objeto, se obtiene el máximo y el mínimo para cada coordenada real entre los puntos pertenecientes al robot. La diferencia entre el máximo y el mínimo es el tamaño máximo del objeto en cada coordenada. Sin embargo, tan sólo dos de esas tres medidas son significativas; debido a la propia naturaleza del láser no se obtienen medidas de profundidad válidas. El valor calculado para el tamaño del objeto en esa dimensión se corresponde con la diferencia de profundidad en la cara visible; es decir, la orientada al láser. Como tamaño característico del objeto se toma la mayor de las dos dimensiones significativas, puesto que sólo se utiliza un valor en el proceso de identificación. El proceso global de obtención de las características reales a partir del láser se describe en la Figura 7.

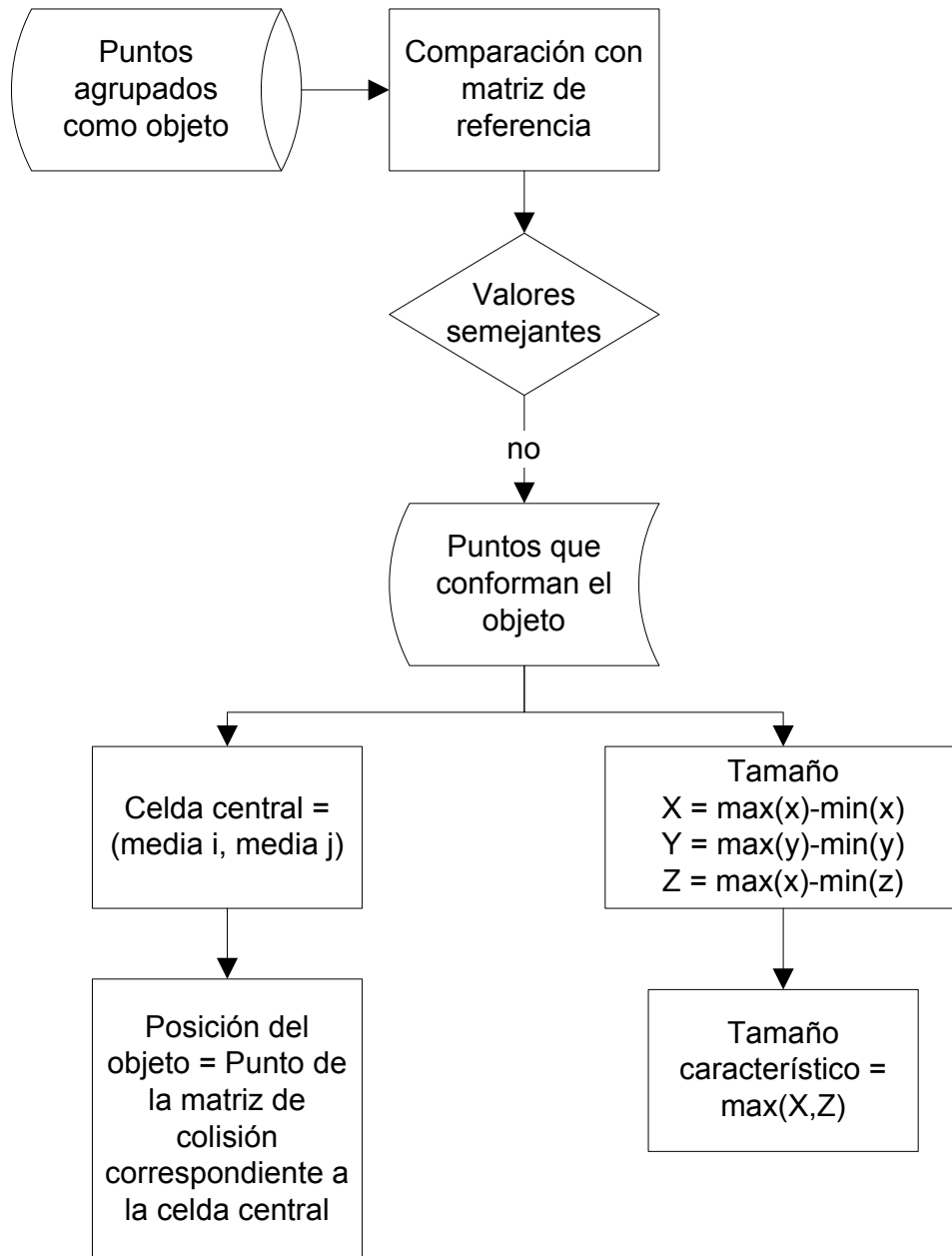


Figura 7: Esquema para la obtención de características reales de los objetos dinámicos

4. ALGORITMOS DE IDENTIFICACIÓN

Para el desarrollo de este proyecto se han empleado dos métodos para la identificación y el seguimiento de los objetos dinámicos, uno de los cuales se puede considerar una extensión de otro. El primer método considera nada más la información de la posición y la velocidad en el plano para realizar la identificación de los objetos, tomando el tamaño como criterio de discriminación para decidir si una identificación ya realizada es correcta. El segundo método combina el cálculo de la distancia de *Mahalanobis* con varios parámetros entre los objetos detectados y los que se mantienen en la lista, con un seguimiento de todos los objetos realizado con un filtro de Kalman de estado variable.

La identificación de objetos dinámicos se realiza entre dos listas diferentes de objetos. La primera es una lista que se realiza con los objetos detectados en un instante determinado. En ella se incluye información de posición, tamaño, color y temperatura que son las características que se pueden extraer de un instante determinado. La segunda lista se mantiene de iteración en iteración para realizar un seguimiento de todos los objetos. El principal problema que se presenta es la asignación de los datos extraídos de los sensores a cada uno de los objetos mantenidos en la lista. Así mismo hay que tener en cuenta algunas excepciones que hacen que el emparejamiento no sea tan simple. El número de objetos en las dos listas no tiene porqué ser el mismo, de hecho en la mayoría de los casos no lo será. Las excepciones más comunes son:

- Detección de un nuevo objeto. Alguno de los objetos detectados en el último instante pueden no encontrarse todavía en la lista permanente puesto que es la primera vez que se detecta.
- Desaparición de un objeto. Tanto por una oclusión parcial o total, porque el objeto detenga su movimiento o porque salga del rango de detección del robot, algunos de los objetos incluidos en la lista permanente pueden no aparecer en la lista de objetos detectados en el último instante.

Se deben también incluir mecanismos de eliminación de objetos móviles para evitar que la lista aumente de tamaño sin ningún control.

4.1. Algoritmo de identificación por mínima distancia.

El primer algoritmo diseñado para el emparejamiento de los objetos entre las dos listas utiliza solamente la información de la posición y la velocidad calculada con las dos últimas posiciones conocidas para cada objeto. La identificación se realiza comparando la posición de los objetos detectados con la posición predicha para los objetos de la lista permanente del robot. Una vez realizada la comparación se eligen las parejas cuya distancia es menor y se comprueba que esa distancia no es demasiado grande. El proceso seguido se describe a continuación:

- Se calcula la posición predicha para cada uno de los objetos de la lista local del robot, utilizando la velocidad y el tiempo transcurrido entre la última vez que se “vio” el objeto y el instante actual, para cada uno de los robots.

$$x_{pred} = x_{k-1} + v_{k-1}^x \cdot \Delta t_k$$

$$y_{pred} = y_{k-1} + v_{k-1}^y \cdot \Delta t_k$$

- Se calcula la distancia entre cada una de las posiciones de los objetos detectados y la posición predicha para los objetos de la lista local, obteniéndose una lista de $n_{det} \times n_{rob}$ parejas, donde n_{det} es el número de objetos detectados en el instante actual y n_{rob} el número de objetos incluidos en la lista local del robot.

$$dist_{ij} = \sqrt{(x_{pred}^i - x_k^j)^2 + (y_{pred}^i - y_k^j)^2}$$

- Se ordenan las parejas de menor a mayor en función de la distancia.
- A continuación se seleccionarán tantas parejas como objetos contenga la lista con menor número de elementos.
- Las parejas se irán seleccionando tomando aquellas cuya distancia entre los objetos que las forman sea menor, evitando en las sucesivas parejas la inclusión de objetos que ya hayan sido seleccionados previamente.

Tras completar la selección de las parejas se debe comprobar que la distancia calculada para cada una de ellas es admisible; es decir, que está por debajo de una tolerancia que se ha de establecer previamente. Este valor puede variar en función de varios parámetros que dependen de la aplicación donde se vaya a insertar el módulo de detección de objetos dinámicos. Influyen en la elección de este parámetro el tipo de objetos que se piense que se va a detectar, la velocidad a la que se espera que se muevan y su tamaño aproximado.

Una vez determinadas las parejas admisibles dentro de la lista de parejas seleccionadas se debe actualizar los datos de los objetos incluidos en estas parejas con la nueva información obtenida. También se ha de calcular la nueva velocidad del objeto; usando para ello la posición anterior, la nueva posición obtenida del análisis de los datos del láser y el incremento de tiempo transcurrido entre la última vez que se observó el objeto y el tiempo actual. Los datos que se actualizan son: posición, tamaño y tiempo de detección; así como la velocidad, aunque este dato es calculado no extraído directamente de los sensores.

El último paso en la actualización de la lista de objetos con la nueva información comprende dos fases; la eliminación de los objetos que llevan demasiado tiempo sin ser detectados y la inclusión de los nuevos objetos observados.

La eliminación de objetos consiste simplemente en recorrer la lista de todos los objetos incluidos actualmente en la lista y eliminar de ella todos aquellos cuyo tiempo de detección no sea lo suficientemente cercano al tiempo actual. Este umbral de tiempo también ha de establecerse previamente, en función de las necesidades que se prevean para el entorno en el que se vaya a utilizar.

La inclusión de objetos en la lista tampoco presenta ningún problema puesto que consiste simplemente en copiar los datos de todos los objetos no emparejados en la lista local del robot poniendo a cero la velocidad. El único parámetro al que hay que asignarle un valor, que no se obtiene en la detección es el número de identificador del objeto que ha de ser único para cada objeto en toda la misión.

El esquema global del algoritmo puede observarse en la

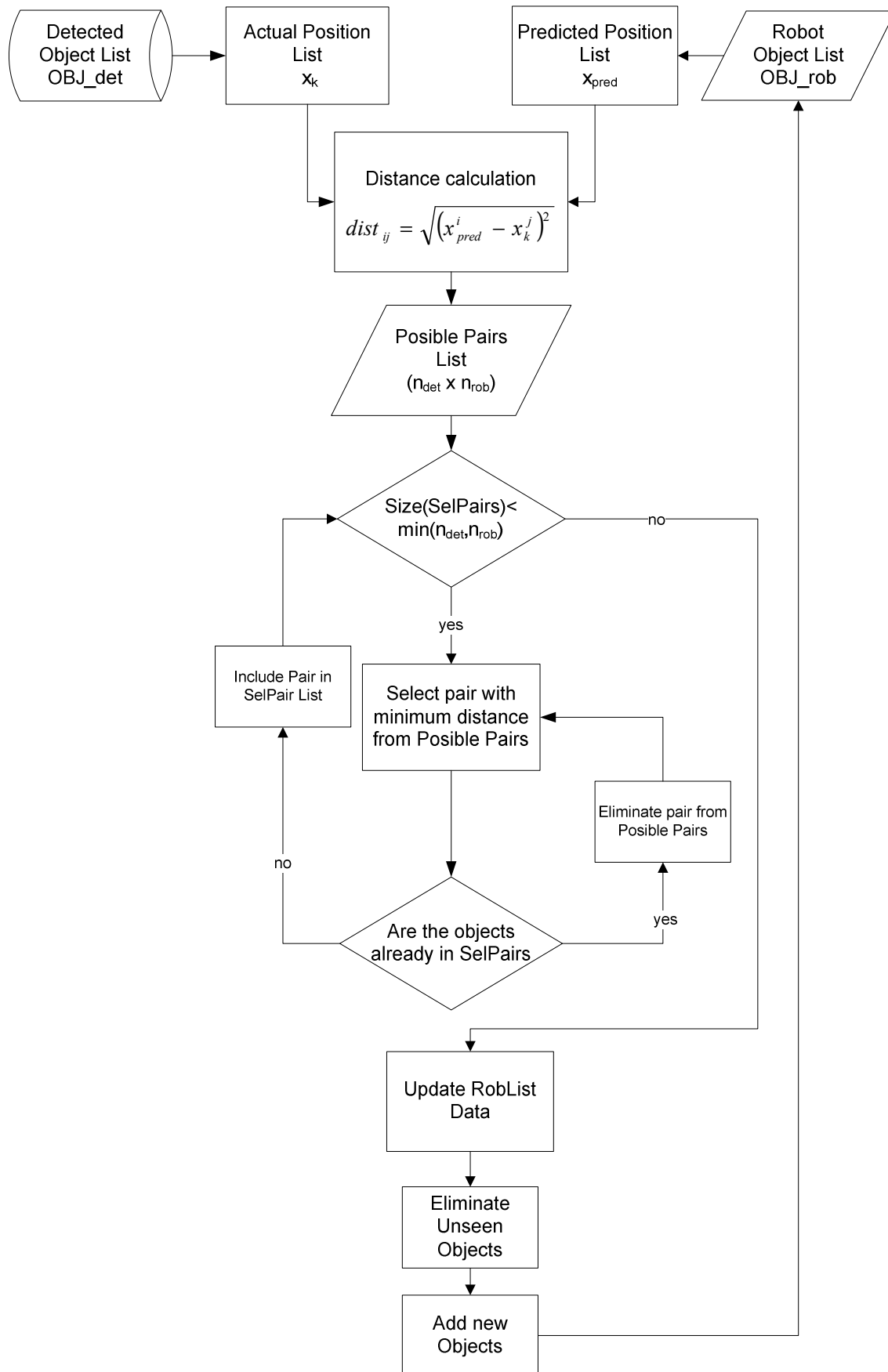


Figura 8.

Gestión de identificadores numéricos

Debido a la posibilidad de un gran número de objetos que pueden aparecer y desaparecer dentro de la misma misión se debe poder identificar cada uno de ellos con un número exclusivo y único en todo el desarrollo de la misión. Para este procedimiento de “etiquetado” se ha diseñado un método muy simple pero muy efectivo que garantiza que en ningún momento se use un identificador ya utilizado previamente.

Se debe guardar en todo momento el máximo identificador asignado a cualquier objeto. Para ello se hace una doble comprobación tanto en el proceso de inclusión de nuevos objetos como en el proceso de actualización de características de las parejas seleccionadas. Cuando se va recorriendo la lista de parejas seleccionadas, se va comparando el valor almacenado del máximo identificador con los identificadores de todos los objetos incluidos en las parejas, manteniendo el máximo de todos ellos. Este procedimiento se repite cuando se comprueban los objetos no detectados en la iteración actual, volviendo a mantener el máximo valor encontrado.

De esta forma, al incluir nuevos objetos se puede garantizar que el valor utilizado para los identificadores es único y no se había empleado anteriormente. La doble comprobación parece excesiva si se tiene en cuenta que se almacena el valor del máximo identificador después de haber incluido los nuevos objetos, pero es necesario debido a las fusiones que se realizan entre la lista local del robot y otras listas de otros robots o la lista global del servidor central en el caso de control centralizado.

Puede ocurrir que el mismo objeto vuelva a aparecer después de haber estado fuera del rango de detección durante mucho tiempo y en este caso se identificaría como un nuevo objeto. Esto podría identificarse como un inconveniente pero no es un problema puesto que la mayoría de las aplicaciones, al cabo de un determinado tiempo, que vuelva a aparecer el mismo objeto pero se lo identifique como uno nuevo sólo implica un nuevo análisis de la situación que habría que realizar en cualquier caso aunque se hubiese identificado como el mismo objeto.

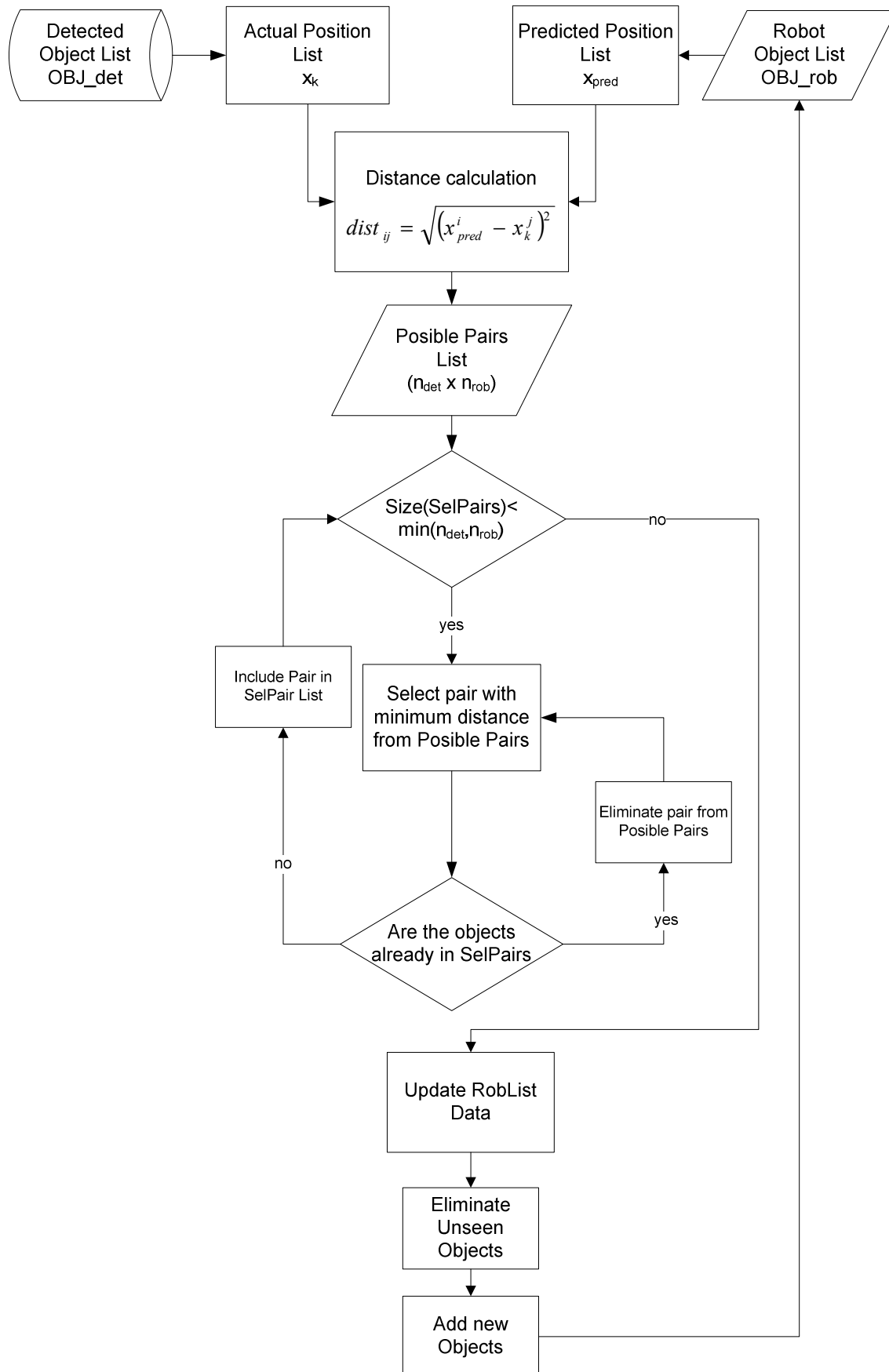


Figura 8: Esquema del algoritmo de identificación de objetos

4.2. Algoritmo de filtro de Kalman de estado variable.

El principal problema del anterior algoritmo es que sólo incluye la información de posición en la identificación de los objetos. El resto de las características del objeto como tamaño, color o temperatura no se utiliza para identificar o predecir el siguiente estado, simplemente se guarda para el uso del operador humano que controla la aplicación. Este desaprovechamiento de información en la identificación de los objetos propició la creación de otro algoritmo que integrase todas las características extraídas de los sensores en un solo proceso de identificación.

Se estudiaron varias alternativas, como la utilización de redes neuronales [31] o filtros de partículas [32]. Sin embargo la diversa procedencia de la información que se ha de integrar propició la utilización de un método capaz de manejar datos de diversas fuentes; en este caso, diseñar un filtro de Kalman que pueda combinar todos los datos. En este caso se diseñará un filtro que maneje datos de posición, tamaño, color y temperatura, además de la velocidad calculada a partir de dos estados consecutivos; pero se pretende que el algoritmo sea extensible y suficientemente genérico para que posteriormente se puedan introducir más variables en el caso de disponer de más información o de más sensores.

El filtro de Kalman es un algoritmo recursivo óptimo para procesar información. Combina la totalidad de la información disponible, ponderándola según su grado de incertidumbre, para realizar la estimación de las variables que definan el estado del sistema. El funcionamiento del filtro requiere el conocimiento de la dinámica del sistema, así como de los modelos estadísticos del ruido en las medidas de los sensores y de la incertidumbre inicial del modelo del sistema. Al tratarse de un algoritmo recursivo, cada estimación se efectúa a partir de la anterior y de la nueva información disponible, sin que sea preciso almacenar todos los datos previos.

El filtro de Kalman permite minimizar el error en la estimación de las variables de interés cuando el modelo es lineal y la incertidumbre del sistema y de las medidas de los sensores es ruido blanco gaussiano. En esta situación, la función de densidad de probabilidad de cada variable a analizar condicionada a las medidas tomadas es tal que la media, la moda y la mediana coinciden, lo que evita cualquier posible conflicto a la hora de determinar cuál es la mejor estimación. Las hipótesis aceptadas pueden parecer altamente restrictivas pero hacen posible la resolución matemática del problema y se acercan bastante bien a la realidad en la mayoría de los casos.

En el caso del manejo de objetos móviles el principal problema que se presenta es la variabilidad del número de variables a tratar; en este caso el número de objetos dinámicos que se presentan dentro del rango del sensor, un número que no se puede

prever previamente a la puesta en marcha del robot. Por lo tanto se ha de usar un filtro de Kalman de estado variable, que se describirá con más detalle más adelante.

La formulación normal del filtro de Kalman [33] consiste en un modelo lineal dinámico del sistema en el que el estado actual k evoluciona del estado anterior:

$$x_k = A_k \cdot x_{k-1} + B_k \cdot u_k + w_k$$

donde x_k es el estado actual, u_k las entradas al sistema y w_k el ruido del sistema que se supone que sigue una distribución normal $N(0, Q_k)$.

En un instante se supone que la observación z_k del estado actual x_k se modela como:

$$z_k = H_k x_k + v_k$$

donde H_k es el modelo de observación y v_k es el ruido de la observación que sigue una distribución normal $N(0, R_k)$.

La predicción del estado del sistema se expresa como:

$$\hat{x}_{k|k-1} = A_k \cdot \hat{x}_{k-1|k-1} + B_k u_k$$

$$P_{k|k-1} = A_k P_{k-1|k-1} A_k^T + Q_k$$

donde $\hat{x}_{k|k-1}$ es la estimación del estado en el instante k considerando las medidas hasta el instante $k-1$ y $P_{k|k-1}$ es la matriz de covarianzas; una medida de la precisión de la estimación del estado.

Con la predicción se puede calcular la innovación y la estimación del estado en el instante actual:

Innovación en la medida: $\tilde{y}_k = z_k - H_k \cdot \hat{x}_{k|k-1}$

Innovación en la covarianza: $S_k = H_k P_{k|k-1} H_k^T + R_k$

Ganancia de Kalman: $K_k = P_{k|k-1} H_k^T S_k^{-1}$

Estimación del estado actualizada: $\hat{x}_{k|k} = \hat{x}_{k|k-1} + K_k \tilde{y}_k$

Estimación de la covarianza actualizada: $P_{k|k} = (I - K_k H_k) P_{k|k-1}$

La aplicación de este algoritmo al problema de la detección de objetos dinámicos no es inmediata. Existen algunos problemas para lo que hay que encontrar solución, el más importante de los cuales es la variabilidad del número de estados, ya que depende del número de objetos detectados en cada instante. A continuación se va a describir cada uno de los parámetros del filtro y los procesos de inclusión y eliminación de objetos en cada iteración.

Estado del sistema x_k :

En este caso el estado lo componen los estados parciales de cada uno de los objetos. Las características que se incluyen en el estado son posición, tamaño, color y temperatura. El estado de un objeto sería:

$$x_k^i = \begin{bmatrix} x \\ y \\ size \\ R_{RGB} \\ G_{RGB} \\ B_{RGB} \\ T \end{bmatrix}$$

Por lo tanto el estado del sistema, sería un vector de $7 \times n$ elementos, donde n sería el número de objetos detectados en el instante actual:

$$x_k = \begin{bmatrix} x_k^1 \\ x_k^2 \\ x_k^3 \\ \vdots \\ x_k^n \end{bmatrix}$$

Además de las características físicas de cada objeto hay que mantener un registro del último instante en que cada objeto fue observado. Este valor se utilizará para ir eliminando objetos de la lista cuando haya pasado demasiado tiempo desde su última detección.

Entradas al sistema u_k :

Las entradas al sistema son solamente dos, las velocidades en el plano. De nuevo habrá tantas parejas de velocidades como objetos se detecten en cada instante.

$$u_k^i = \begin{bmatrix} v_x \\ v_y \end{bmatrix} \quad u_k = \begin{bmatrix} u_k^1 \\ \vdots \\ u_k^n \end{bmatrix}$$

Las velocidades se inicializarán a un valor cero cuando se detecta un objeto por primera vez. Para las siguientes iteraciones el valor de la velocidad se calcula a partir del estado actual y el estado anterior del sistema, tomando en consideración el tiempo transcurrido entre los dos estados.

Modelo del sistema A_k y B_k :

Se trata de un modelo lineal en el que se considera que los objetos se mueven a velocidad constante mientras el resto de las características permanecen invariantes puesto que es lógico pensar que ni el tamaño, el color o la temperatura van a cambiar bruscamente. Se pueden producir cambios en el tamaño debido a un cambio de orientación del objeto o un cambio de temperatura debido a la acción del sol o a un sobrecalentamiento interno. También pueden producirse cambios en el color si las condiciones lumínicas cambian o si el objeto emplea alguna técnica de camuflaje. En cualquier caso, la posibilidad de que todas las características del objeto cambien bruscamente a la vez es muy improbable. El filtro de Kalman está diseñado para asimilar estos cambios y reaccionar ante ellos.

Por lo tanto las matrices del modelo son:

$$A_k^i = \begin{bmatrix} 1 & 0 & & & 0 \\ 0 & 1 & & & \\ & & 1 & & \\ \vdots & & \ddots & 1 & \ddots \\ & & & 1 & \\ 0 & & & & 1 & 0 \\ & & & & 0 & 1 \end{bmatrix} \quad y \quad B_k^i = \begin{bmatrix} \Delta t & 0 \\ 0 & \Delta t \\ 0 & 0 \\ 0 & 0 \\ 0 & 0 \\ 0 & 0 \end{bmatrix}$$

En cuanto a las matrices globales del modelo:

$$A_k = \begin{bmatrix} A_k^1 & 0 & 0 \\ 0 & \ddots & 0 \\ 0 & 0 & A_k^n \end{bmatrix} \quad y \quad B_k = \begin{bmatrix} B_k^1 & 0 & 0 \\ 0 & \ddots & 0 \\ 0 & 0 & B_k^n \end{bmatrix}$$

Valores iniciales de covarianzas y errores P_{ini} , Q_{ini} y R_{ini}

La matriz de covarianza inicial es una medida de la bondad de la estimación inicial de cada una de las características de un objeto con respecto a su valor real. Para cada objeto, su matriz de covarianzas inicial puede escribirse como:

$$P_{ini} = \begin{bmatrix} P_{pos} & 0 & & & 0 \\ 0 & P_{pos} & & & \\ & & P_{size} & & \\ \vdots & & \ddots & P_{RGB} & \ddots \\ & & & P_{RGB} & \\ & & & & P_{RGB} & 0 \\ 0 & & & & 0 & P_{Temp} \end{bmatrix}$$

donde los valores P_{pos} , P_{size} , P_{RGB} y P_{Temp} se eligen al comienzo de cada misión en función de la fiabilidad de cada uno de los sensores. En el caso de que no se conozca con seguridad la bondad de los sensores es más correcto utilizar un valor alto para las covarianzas iniciales ya que el filtro de Kalman está diseñado para manejar estas incertidumbres y mejorar las estimaciones con cada iteración. Sin embargo si se emplea un valor pequeño de las covarianzas (se estima que las medidas iniciales son buenas) y las medidas iniciales no se aproximan a los valores reales esto puede provocar que el filtro no sea capaz de converger a las solución correcta.

Las varianzas en el ruido de la medida (R_{ini}) y las varianzas en el ruido del proceso (Q_{ini}) son matrices de igual dimensión que P_{ini} en la que los elementos de la diagonal indican los valores de R y Q para cada uno de los parámetros que se utilizan en el modelo. Los valores de R expresan la bondad de las medidas de los sensores y se han de elegir en función de la sensibilidad de cada uno. El ruido del proceso (Q) representa la variación entre el valor real de las características para cada objeto y la estimación obtenida a través de los sensores.

Valores obtenidos de los sensores. Medidas.

Las medidas obtenidas de los sensores coinciden con los parámetros que se incluyen en el vector de estado aunque no se puede saber a priori que conjunto de medidas corresponden a que objeto. Para cada objeto detectado, el vector de medidas incluye:

$$z_k^i = \begin{bmatrix} z_x \\ z_y \\ z_{size} \\ z_R \\ z_G \\ z_B \\ z_T \end{bmatrix}$$

Inicialización del filtro de Kalman

Al comienzo de la detección de los objetos dinámicos es necesario inicializar todas las matrices del filtro. Con la lista de los objetos inicial se han de crear todas las matrices necesarias para que en la iteración siguiente el filtro pueda ejecutarse de forma normal. Estas matrices son A_k , B_k , P_k , x_k , Q_k , R_k , U_k y en cada una de ellas se debe colocar un valor específico de inicialización. Para las matrices A_k y B_k estos valores son los determinados por el modelo del sistema y el número de objetos detectados por primera vez. P_k , Q_k y R_k se inicializarán con unos valores propios para cada aplicación. Para x_k se utilizarán las características de los objetos detectados en esta primera iteración; mientras que las velocidades de entrada U_k se inicializarán a cero puesto que no se conoce ninguna información adicional de la posición anterior de los objetos.

4.2.1. Iteración del filtro de Kalman. Adición y extracción de objetos.

El primer paso al recibir una nueva lista de objetos del módulo de detección es realizar la predicción del estado para el instante siguiente.

$$\hat{x}_{k|k-1} = A_k \cdot \hat{x}_{k-1|k-1} + B_k u_k$$

$$P_{k|k-1} = A_k P_{k-1|k-1} A_k^T + Q_k$$

Una vez realizada la predicción se debe establecer la correspondencia entre los objetos incluidos en el estado del sistema y los objetos de la lista de objetos dinámicos proporcionados por el módulo de detección. Para ello se emplea la distancia de Mahalanobis, que es un método para determinar la similitud entre dos variables aleatorias multidimensionales. Formalmente se puede expresar como:

$$\vec{d}(x, y) = \sqrt{\sum_{i=1}^N \frac{(x_i - y_i)^2}{\sigma_i^2}}$$

El emparejamiento se realiza siguiendo un proceso similar al del algoritmo anterior. Primero se calcula la distancia de Mahalanobis para cada una de las posibles parejas de objetos entre las dos listas. Una vez se dispone de la lista con todas las parejas se han de seleccionar el número máximo de parejas posibles; es decir, tantas como objetos tenga la lista con menor número de elementos. La selección de las parejas en la lista se realiza seleccionando aquellas en las que la distancia de Mahalanobis es menor, siempre y cuando ninguno de los objetos esté en una pareja previamente seleccionada. Las parejas con menor distancia de Mahalanobis serán aquellas con características más similares, no sólo en la posición sino también en parámetros como el color, el tamaño o la temperatura.

Cuando se obtiene la lista de las parejas seleccionadas se puede realizar la iteración en el filtro. Para ello se debe crear una serie de variables auxiliares en las que sólo se incluyen los datos de los objetos que forman parte de las parejas seleccionadas; es decir, el vector de estado y las matrices necesarias para realizar la iteración del filtro. Estas variables son:

- $\hat{x}_{k|k-1}^{sub}$, subconjunto de $\hat{x}_{k|k-1}$ donde se incluyen todos los objetos de la lista de parejas seleccionadas.
- z_k^{sub} , subconjunto de z_k donde se incluyen las medidas correspondientes a los objetos de la lista de parejas seleccionadas en el mismo orden en que aparecen en $\hat{x}_{k|k-1}^{sub}$.
- $P_{k|k-1}^{sub}, H_k^{sub}$ y R_k^{sub} , subconjuntos de $P_{k|k-1}, H_k$ y R_k respectivamente respetando igualmente el orden de los objetos utilizado en las anteriores variables.

Tras la obtención de los subconjuntos de las variables correspondientes, se puede proceder al cálculo de la estimación del nuevo valor del vector de estado y de la matriz de covarianzas.

Innovación en la medida:

$$\tilde{y}_k = z_k^{sub} - H_k^{sub} \cdot \hat{x}_{k|k-1}^{sub}$$

Innovación en la covarianza:

$$S_k = H_k^{sub} P_{k|k-1}^{sub} (H_k^{sub})^T + R_k^{sub}$$

Ganancia de Kalman:

$$K_k = P_{k|k-1}^{sub} (H_k^{sub})^T S_k^{-1}$$

Estimación del estado actualizada:

$$\hat{x}_{k|k}^{sub} = \hat{x}_{k|k-1}^{sub} + K_k \tilde{y}_k$$

Estimación de la covarianza actualizada:

$$P_{k|k}^{sub} = (I - K_k H_k^{sub}) P_{k|k-1}^{sub}$$

Para obtener el vector de estado completo se deben incluir en el vector de estado los objetos no emparejados de cada una de las listas. Primero, del vector de estado en instante anterior, se deben incluir los objetos para los que no se han obtenido medidas. En segundo lugar, de todos los objetos incluidos en el vector de medidas z_k , se deben incluir aquellos que previamente no estaban en el vector de estado. En resumen, se deben incluir los objetos presentes en $\hat{x}_{k|k-1}$ y en z_k que no hayan sido emparejados. Para los objetos de los cuales no se han obtenido medidas los valores que se incluyen en el vector de estado son las predicciones incluidas en $\hat{x}_{k|k-1}$ sin las correcciones proporcionadas por el filtro de Kalman. Los valores utilizados para los nuevos objetos incluidos en el vector de estado son directamente los extraídos del vector de medidas.

Cálculo del vector de entradas. Velocidad

El cálculo del vector de entradas es muy simple puesto que se limita al cálculo de la velocidad para cada objeto; es decir, el cociente entre la diferencia la posición actual y la anterior y el incremento de tiempo entre las dos medidas. El incremento de tiempo entre las dos medidas se calculará como la diferencia entre el tiempo actual y el tiempo de la última detección para cada objeto. Para los nuevos objetos incluidos en el vector de estado la velocidad empleada será cero puesto que no se conoce una posición anterior.

El principal problema de este proceso no radica en el cálculo de la velocidad sino en emparejar correctamente los datos. En dos iteraciones sucesivas los objetos no están ordenados de igual modo en el vector de estado, por lo que es necesario conocer la correspondencia entre los elementos.

$$U_k^i(v_x) = \frac{x_{k|k}^i[x] - x_{k-1|k-1}^i[x]}{t_k - t_{k-1}}$$

$$U_k^i(v_y) = \frac{x_{k|k}^i[y] - x_{k-1|k-1}^i[y]}{t_k - t_{k-1}}$$

5. SISTEMAS MULTIROBOT. FUSIÓN DE INFORMACIÓN PARA LA IDENTIFICACIÓN DE OBJETOS DINÁMICOS.

Un sistema multirobot es un sistema compuesto por múltiples robots que interactúan entre ellos. Se pueden usar para resolver problemas o ejecutar tareas que serían imposibles o muy difíciles de llevar a cabo de forma individual. Las características principales de un sistema multirobot son la autonomía, ya que cada robot deberá ser capaz de actuar parcial o totalmente de forma autónoma; la visión local, puesto que ningún robot dispone de una visión global del sistema o en algunos casos el sistema al completo es demasiado complejo para que ninguno de los robots pueda hacer un uso efectivo de toda esa información. Otra característica ventajosa de los sistemas multirobot es que pueden manifestar comportamientos complejos aunque sus miembros individuales tengan estrategias de comportamiento simples. Los robots pueden compartir todo tipo de información dentro de las limitaciones del protocolo de comunicación usado en cada caso.

En aplicaciones de vigilancia y seguridad el uso de sistemas multirobot está todavía más justificado debido a la gran extensión de terreno que normalmente se ha de cubrir. Toda esta extensión implica que de usar un solo robot una gran parte del perímetro quedaría sin vigilancia efectiva durante mucho tiempo, o en el caso de que se incluyesen sensores adicionales y se detectase una amenaza el único robot disponible podría tardar mucho tiempo en llegar a la zona afectada o, en función del tipo de estructura de control que se implemente, ni siquiera enterarse de la amenaza hasta mucho tiempo después.

Por lo tanto es importante distinguir entre los dos principales tipos de estructura de control que se usan actualmente, control centralizado y control descentralizado. El control centralizado es un tipo de estructura fuertemente jerárquica en la que existen diferentes niveles en forma de pirámide y en la que el objetivo global de la misión sólo es conocido por el último nivel de la pirámide. En el caso de los sistemas multirobot esto suele implicar que uno de los robots asume el mando y dirige a los demás, o que existe una estación base que recibe la información de todos los robots y les envía las ordenes con los objetivos locales a cada uno así como la información que necesitan para llevar a cabo estas misiones. Esta estación central puede disponer o no de un operador humano dependiendo del tipo de aplicación que se esté desarrollando. La principal ventaja de este tipo de control es la uniformidad de las decisiones; es decir, no se producirán situaciones en las que dos robots intenten llevar a cabo la misma acción por falta de comunicación. Otra ventaja importante es que este sistema

aprovecha de forma óptima las diferentes características especiales de cada robot; es decir, la estación central optimiza el uso de cada uno de los robots. También hay que destacar la economía, puesto que es más barato añadir capacidad de procesamiento a la unidad central que añadirla a cada uno de los elementos del grupo.

La otra forma de control más aplicada en desarrollos de sistemas multirobot es el control descentralizado en la que cada uno de los elementos del sistema toma decisiones de forma autónoma basándose en los datos tomados por sus sensores y en la información recibida de los robots dentro de su rango de comunicación. Cada uno de los robots puede ser independiente o pueden existir jefes de grupo que toman decisiones por los miembros de su grupo. La principal ventaja de este tipo de sistemas es el aumento de confiabilidad que supone no tener una estación central que toma todas las decisiones de importancia, ya que el fallo de cualquier elemento del sistema no implica el fallo del conjunto. Así mismo, la capacidad de crecimiento del sistema es incremental puesto que añadir un elemento al sistema no implica aumentar la capacidad de procesamiento o el tamaño de la estación central; únicamente habría que añadir un nuevo robot al sistema.

En el caso de la detección de objetos dinámicos y de la colaboración entre los distintos robots también se pueden implementar estas dos estructuras de fusión de datos para compartir la información de los posibles intrusos entre todos los robots y poder tomar decisiones con la mayor cantidad de información posible.

Tanto la arquitectura de control centralizada y la descentralizada tienen ventajas e inconvenientes que ya se han mencionado y en la mayor parte de los desarrollos la aplicación de una u otra estructura dependerá en gran parte de la misión que haya que ejecutar en cada momento. Además en algunas misiones habría que combinar las estructuras de control y en otros casos dividir los robots para que unos trabajen de forma centralizada y otros de forma descentralizada. Por lo tanto, en lugar de desarrollar una fusión de información para cada uno de los casos, que podría ser vulnerable a fallos si se combinasen las estructuras de control y que requeriría unos algoritmos muy complejos, se optó por desarrollar un algoritmo simple pero efectivo que fusionase la información en todos los casos. Este algoritmo deberá fusionar la información de los objetos dinámicos del entorno tanto si todos los robots se encuentran controlados por una estación central como si se encuentran trabajando de forma autónoma.

5.1. Algoritmo de fusión de información

Para poder diseñar un algoritmo que fusione de forma efectiva la información referente a los objetos dinámicos hay que tener en cuenta que, independientemente de la arquitectura utilizada, siempre habrá que realizar la fusión entre dos listas distintas. En todos los casos, tanto en la estación central como en uno de los robots, una de las listas será local (L_{loc}), que en el caso de los robots es la que van actualizando a medida que realizan medidas con el láser y demás sensores; y la otra será la lista de entrada (L_{ent}) que es la que viene suministrada por la estación central o por otro robot.

De cualquier forma habrá que fusionar la información contenida en la lista de entrada con la información que dispone la lista local para generar una nueva lista local actualizada con toda la información. El flujo de las listas dependerá de la arquitectura de control utilizada y se describirá más adelante para los dos casos principales (centralizado y descentralizado).

El proceso de fusión de las dos listas se asemeja mucho a la identificación que se realiza cuando se incorporan las nuevas medidas de los objetos a la lista local de cada robot. En primer lugar hay que conocer con exactitud la información que contiene cada lista de objetos. Para cada objeto incluido en la lista se dispone de la siguiente información:

- Posición: Coordenadas en el plano (x,y) de la última posición conocida del objeto.
- Velocidad: Velocidad en el plano del objeto.
- Características físicas: Tamaño característico del objeto, normalmente la mayor dimensión apreciable por los robots.
- Tiempo de detección: Tiempo al que corresponden en resto de las medidas. Este dato es de la mayor importancia puesto que nos permitirá realizar la fusión correctamente.
- Identificador: Número que identifica al objeto. Se trata de un número de 5 cifras en las que las dos primeras identifican al robot que realizó la detección y el resto será el identificador del objeto.

El proceso de fusión de las dos listas sigue el mismo patrón que en el de identificación; en primer lugar hay que calcular las distancia entre todas las posibles parejas de objetos y seleccionar las más cercanas. Después, con las parejas de objetos se actualizará la lista local con la información más actualizada.

La creación de las parejas no es tan sencilla como en el algoritmo de identificación. En este caso hay que tener en cuenta que las posiciones de los objetos han sido obtenidas en tiempos distintos para cada objeto, por lo que antes de calcular la distancia entre cada pareja de objetos hay que extrapolar una de las posiciones hasta el tiempo correspondiente a la medida de la otra posición. Este proceso se explica en profundidad a continuación:

- Se determina el número de objetos de cada lista; n_{loc} para la lista local y n_{ent} para la lista de entrada. De esta forma se van a calcular $n_{loc} \times n_{ent}$ parejas distintas.
- Se toma un objeto de la lista local (i_{loc}) y otro de la lista de entrada (i_{ent}) y se comparan los tiempos de detección. La medida que corresponda a un tiempo de detección mayor; es decir, la medida más reciente, es la que se mantendrá intacta mientras que se calculará la posición predicha para la medida proveniente de la otra lista. Este proceso se lleva a cabo para todas las parejas posibles de objetos entre las dos lista siempre y cuando la diferencia entre los tiempos de detección no supere los 15 segundos.

$$Si \text{ Abs}(t_{det}^{loc} - t_{det}^{ent}) < 15$$

$$t_{det}^{loc} > t_{det}^{ent} \rightarrow x_{pred}^{ent} = x_{ent} + v_{ent} \cdot (t_{det}^{loc} - t_{det}^{ent})$$

$$t_{det}^{loc} < t_{det}^{ent} \rightarrow x_{pred}^{loc} = x_{loc} + v_{loc} \cdot (t_{det}^{ent} - t_{det}^{loc})$$

- Una vez se dispone de dos medidas de posición se calcula la distancia entre los objetos de las dos listas.

$$dist_{i_{loc}i_{ent}} = \sqrt{(x_{pred}^{ent} - x_{loc})^2 + (y_{pred}^{ent} - y_{loc})^2}$$

$$dist_{i_{loc}i_{ent}} = \sqrt{(x_{pred}^{loc} - x_{ent})^2 + (y_{pred}^{loc} - y_{ent})^2}$$

- Se ordenan todas las parejas de menor a mayor en función de la distancia.
- Se seleccionan tantas parejas como objetos contenga la lista con un menor número de elementos, empezando por los de menor distancia.
- Las parejas se seleccionarán tomando aquellas cuya distancia entre los objetos que la forman sea menor, y evitando en sucesivas parejas la inclusión de objetos que ya hayan sido seleccionados previamente. Además se debe comprobar que la distancia calculada para cada una de las parejas elegidas está por debajo de un límite de tolerancia que se establece al inicio de cada misión. En caso contrario esa pareja y todas las siguientes en la lista serán descartadas, puesto que se trata de objetos demasiado alejados entre ellos.

Una vez se dispone de las parejas seleccionadas se debe actualizar la información de la lista local. En primer lugar se toman los objetos de las dos listas que no pertenecen a ninguna de las parejas seleccionadas; los objetos de la lista local se mantienen con la misma información y los objetos de la lista de entrada se incluyen en la lista local asignándoles un nuevo número identificador. Para los objetos de las parejas seleccionadas hay que comprobar que información es la más reciente; es decir, que tiempo de detección es mayor. A la hora de incorporar este objeto a la lista local se incluirá la información más reciente, pero manteniendo en cualquier caso el número identificador original del objeto de la lista local. El proceso se describe en la Figura 9.

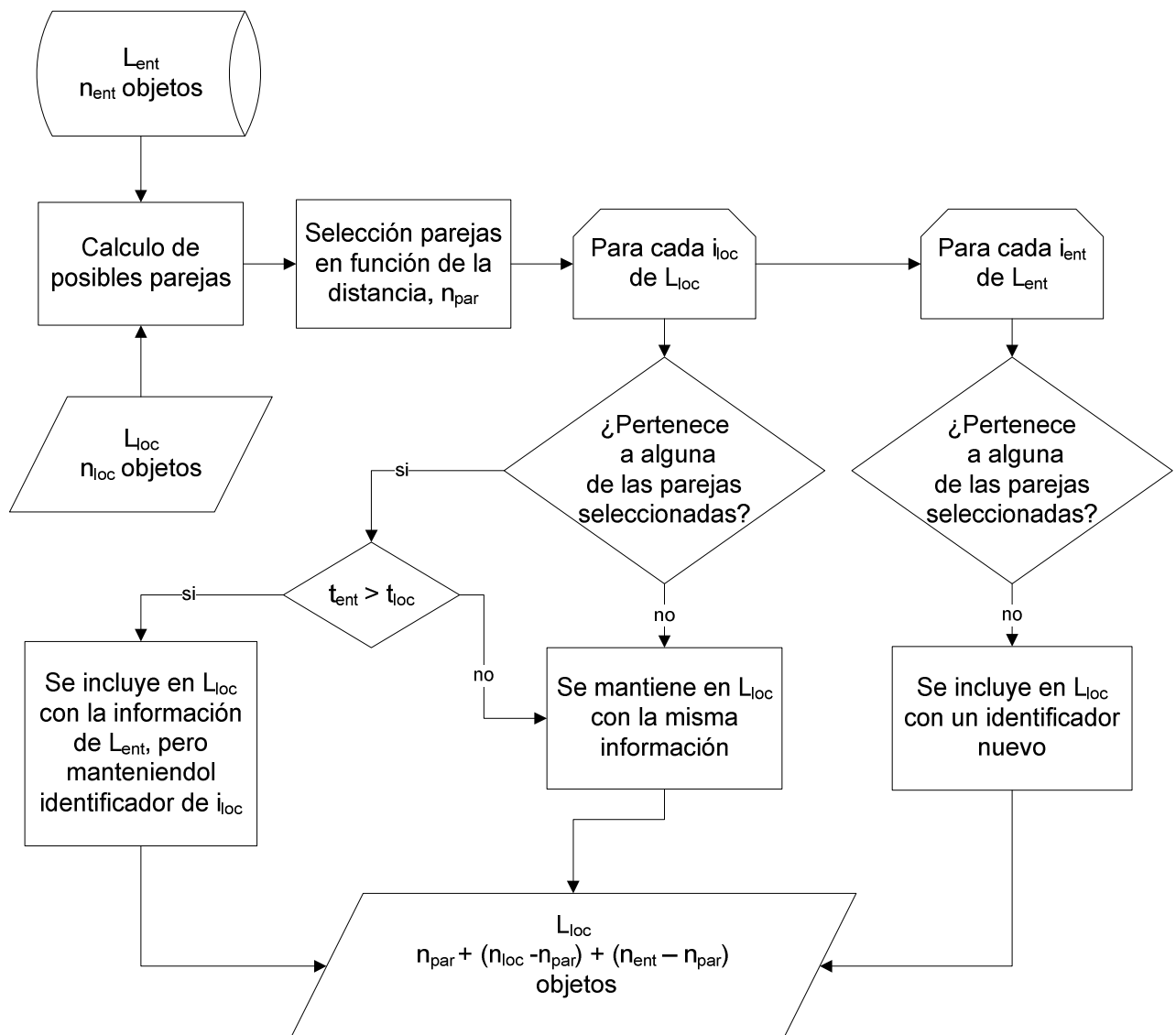


Figura 9: Esquema del algoritmo de fusión de información

5.2. Arquitecturas utilizadas. Ventajas e inconvenientes.

Al implementar este algoritmo en una flota de robots controlados de forma centralizada o descentralizada el flujo de información será diferente en cada caso. Cada sistema tiene una serie de ventajas e inconvenientes que en el caso de la detección de objetos dinámicos afectan sobre todo al tiempo que tarda en llegar la información referente a los objetos a cada robot y a la cantidad de información que le llega. A continuación se describirán brevemente las dos estructuras de control utilizadas y se resaltarán sus ventajas e inconvenientes.

En el caso centralizado, cada robot manda su lista local a la base central que las va fusionando a medida que son recibidas. Con cada nueva fusión, la estación de control genera una nueva lista local que es enviada de nuevo a todos los robots. Cuando un robot recibe una lista de la estación central la fusiona con su lista local e incorpora la información de los objetos móviles observados por el resto de los robots. En la Figura 10 se observa un esquema del flujo de información en esta estructura.

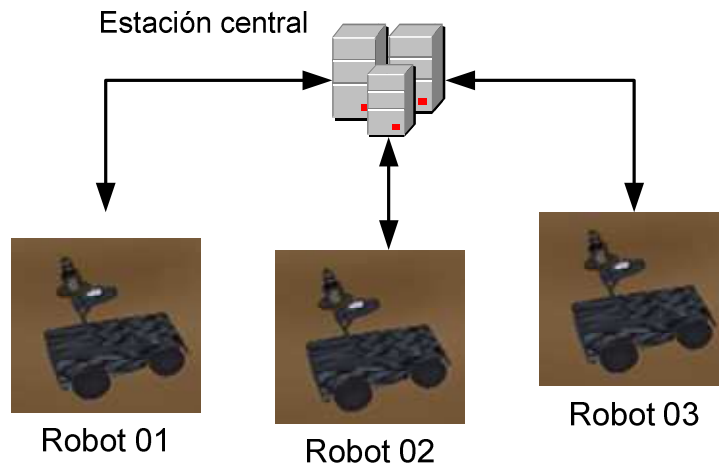


Figura 10: Esquema del flujo de información para estructura centralizada

La principal ventaja que presenta esta estructura es que la información que obtiene cada robot es total; es decir, tiene conocimiento de todos los objetos dinámicos detectados por el resto de los robots. Esto es ventajoso para el desarrollo de la misión puesto que la estación central, o el operador, puede tomar decisiones con la mayor cantidad de información posible, pudiendo dirigir a los robots a las zonas con mayor cantidad de objetos móviles. Sin embargo, esta estructura implica que la información sobre los objetos móviles tarda más en llegar a los robots, ya que ha de mandarse primero a la estación central y una vez fusionada, reenviada al resto de los robots. De esta forma podría darse el caso de que un robot no informara a otro muy

cercano a él de la presencia de un intruso que se dirigiese en su dirección, permitiendo que el primer robot cambiase de dirección y no detectase al intruso.

En el caso descentralizado no existe una estación central que reciba las listas de todos los robots sino que son los propios robots los que se comunican entre sí, enviándose las listas y fusionando la información con la que ellos disponen localmente. En la Figura 11 se observa un esquema de esta arquitectura de comunicación; en el caso de que hubiera más robots se conectarían a todos los existentes en la red.

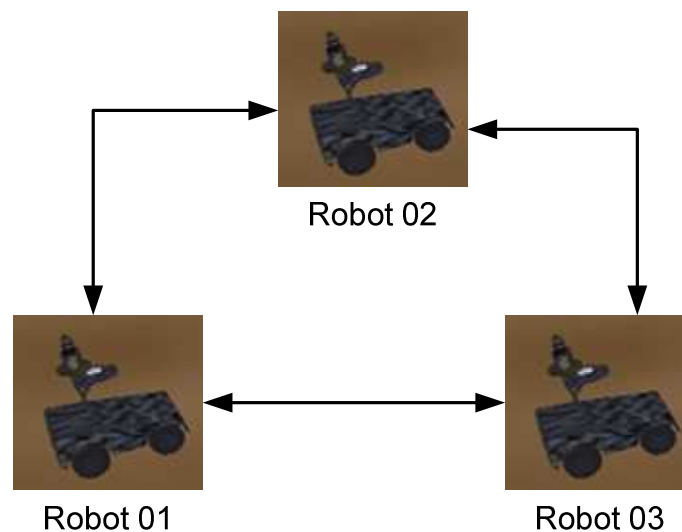


Figura 11: Esquema del flujo de información para estructura descentralizada

La principal ventaja de este sistema es que los robots reciben información de su entorno más rápido que si tuviesen que enviar las listas de objetos dinámicos a la estación central. Esto ayuda a que los robots puedan decidir en qué dirección moverse, tanto para obtener una mejor visión de un intruso como para poder evitar un posible peligro que se dirija a su dirección. Esta forma de enviar la información presenta un gran inconveniente derivado del rango de comunicación de cada robot; y es que cada robot sólo recibe información de los robots que estén en su zona de influencia. Esto implica que el robot no tendrá información de todo el escenario de la misión sino sólo de los objetos detectados por los robots cercanos a él. Obviamente, a medida que avance la misión y los robots vayan comunicándose entre sí cada robot podrá tener conocimiento sobre objetos dinámicos muy alejados de él, pero la información que reciba estará desactualizada debido al retraso que supone tener que esperar a que varios robots reciban información de sus vecinos, la fusionen con la suya propia y la reenvíen.

6. DETECCIÓN DE OBJETOS MÓVILES DESDE UNA PLATAFORMA EN MOVIMIENTO.

En el caso de tomar las medidas del láser desde una plataforma en movimiento el proceso de detección de objetos móviles tiene que ser radicalmente distinto. La diferenciación directa entre dos matrices del láser deja de ser un enfoque válido puesto que, aun no existiendo movimiento delante del robot, las matrices del láser cambian conforme el robot se está moviendo. La principal dificultad radica en separar los cambios en la matriz producidos por el movimiento del robot y los cambios producidos por objetos móviles.

Otra dificultad que presenta la obtención de datos con un sistema LIDAR situado en una plataforma en movimiento es el tiempo total de realización de la medida. Debido a las limitaciones físicas del escáner y sobre todo a las del servo que permite el giro vertical del escáner, todas las distancias no se miden en el mismo instante de tiempo. Aunque se ha supuesto que todas las medidas se realizan en el mismo instante esto puede inducir a pequeños errores si los objetos en movimiento tienen poca velocidad. Una posible solución para este problema es la utilización de sistemas LADAR, que obtienen una imagen 3D con la emisión de un solo pulso láser.

Para abordar todos estos problemas y encontrar una solución factible se ha optado por comparar dos reconstrucciones sucesivas del terreno obtenidas a partir de dos matrices de láser consecutivas. Estas reconstrucciones son mapas de celdas de varias capas en las que cada celda tiene asociado una serie de valores numéricos calculados en base a los puntos de colisión del láser cuya proyección en el plano se encuentra en esa celda. Para cada celda se obtiene el valor de la confianza, la altura media y el valor del gradiente en X y en Y. El proceso de transformación de los puntos de la matriz a puntos de colisión en el sistema de coordenadas global ya ha sido descrito en un apartado anterior por lo que no se incidirá en ello.

La reconstrucción de las elevaciones y depresiones del terreno es parte fundamental del proyecto, siendo datos de partida para el cálculo posterior de los gradientes y con ellos el nivel de ocupación o de accesibilidad del terreno. No se hace ninguna distinción entre terreno y otros obstáculos presentes como edificios, estructuras, vehículos, otros robots, etc. Cada celda del mapa tiene asociado un valor numérico de la altura, que corresponderá básicamente con la altura media de todos los puntos observados en el interior de ella, ponderando dicha información con la estimación de la precisión en la observación de dicho punto de colisión. Para identificar los objetos móviles se almacenará la altura máxima de la celda en lugar de

la altura media puesto que es un indicador más fiable para detectar el movimiento en el mapa.

Las capas de gradientes tienen como objeto almacenar la estimación de las pendientes del terreno para cada celda según orientaciones de los ejes principales X e Y, Los gradientes son calculados siempre respecto a ejes fijos. Así, el gradiente según X mide la variación de la altura según avanzamos en el sentido marcado por el eje X fijo. En esta capa se almacena el valor de la tangente del ángulo que forma la superficie con la horizontal. Es importante hacer notar que el gradiente es calculado utilizando para ello las alturas de los puntos de colisión que se corresponden a la misma celda, es decir, no se realiza un gradiente entre celdas adyacentes, sino en el interior de la misma. Aunque en general los resultados no deben variar demasiado, se ha elegido esta forma de actuar por ser más precisa ya que representa el gradiente justo en la celda en la que se produce el cambio de altura, y si se hiciera un gradiente entre celdas, este se encontraría algo desplazado respecto de la variación de altura producida.

La capa de confianza se utiliza para tener una medida de lo fiables que son los datos almacenados en cada celda. Puesto que la reconstrucción se realiza a partir de las medidas de distancias obtenidas con un láser de barrido horizontal y vertical con incrementos de paso muy pequeños, se obtienen un gran número de puntos de colisión. Cada punto de colisión es localizado en el mapa y asignado a su correspondiente celda, es posible por lo tanto que haya celdas que recojan un gran número de puntos de colisión y otras con un número pequeño de puntos, o incluso ningún impacto. De esta forma, con la confianza solo se registra la información proporcionada por los puntos donde el láser ha impactado, ignorando el resto de puntos por donde el láser ha “sobrevolado”. Otro efecto a considerar es que para una misma celda, puede ocurrir que los puntos estén muy juntos en una pequeña zona, o que por el contrario estén separados ocupando gran cantidad de la superficie de la misma. Es lógico plantearse qué aporta más información, si unas celdas donde el número de puntos sea reducido pero su dispersión sea grande como puede ocurrir en las celdas (1,2) y (2,3) de la Figura 12 si por el contrario aporta más información la situación de la celda (1,1) donde los puntos no presentan demasiada dispersión pero existe gran cantidad de puntos de impacto.

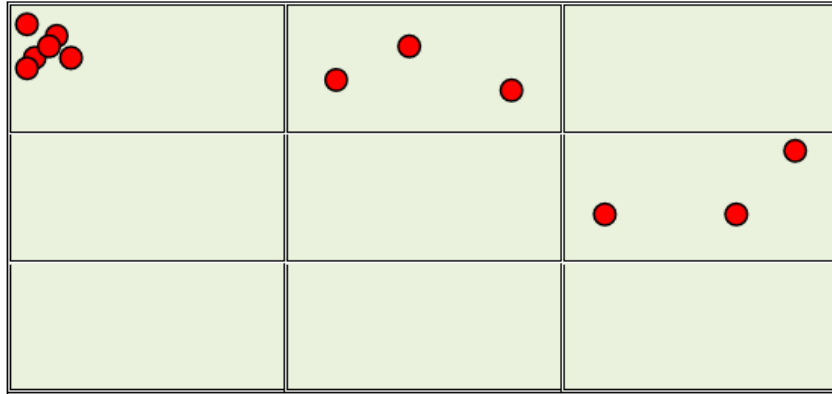


Figura 12: Ejemplo de mapa de puntos de colisión

Por ese motivo, además del número de puntos de impacto sobre una celda hay que tener en cuenta la dispersión de esos puntos, así surgieron durante el desarrollo del software dos tipos de algoritmos que fueron implementados:

- Confianza basada en el número de puntos de impacto
- Confianza basada en la dispersión de los puntos de impacto.

De estos dos tipos de algoritmos investigados que se desarrollaran en apartados posteriores, el segundo de ellos resultó ser más eficaz y fue el elegido para la implementación definitiva.

A medida que el robot se va desplazando por el mapa, puede ocurrir que algunos de los nuevos puntos de colisión no queden dentro de la zona definida por el mapa de celdas. Por lo tanto, antes de realizar la reconstrucción de cada medida del láser es necesario comprobar que el máximo alcance del robot no va a producir puntos fuera del mapa. La condición necesaria para que esto se cumpla es que el robot esté a menos distancia de los límites del mapa que el rango del láser, que en el entorno de simulación utilizado es de 80m. En la Figura 13 se observa cómo se desplaza el mapa a medida que el robot se mueve.

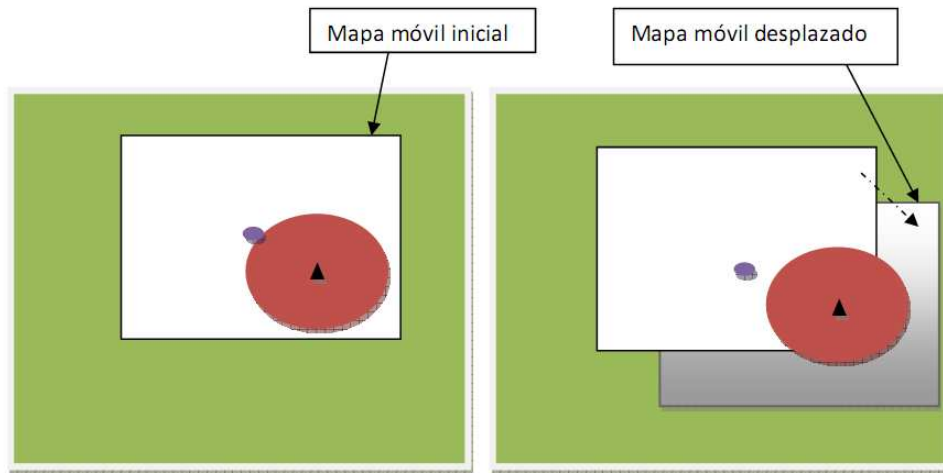


Figura 13: Ejemplo de desplazamiento del mapa a medida que se mueve el robot

A continuación se presenta un esquema de todo el algoritmo de detección de movimiento con el robot en movimiento. Para describirlo completamente hay que tener en cuenta que se usan 3 reconstrucciones consecutivas para asegurar que los cambios de alturas no se deben a objetos inmóviles a los que el robot se está acercando, y por tanto tomando mejores medidas de ellos.

En primer lugar, con dos reconstrucciones sucesivas (R_k y R_{k-1}), siempre que tengan el mismo centro, se realiza un mapa de movimiento (M_k) siguiendo las siguientes condiciones:

- La confianza de cada celda debe ser mayor que un valor mínimo en los dos mapas.
- La diferencia de alturas máximas entre las dos celdas debe ser mayor que un valor que dependerá de la misión y de la sensibilidad que se desee. En este caso se usarán 0,5m.
- La diferencia entre los gradientes en X y en Y debe ser mayor que cero y menor que un valor que también dependerá de la misión. Estos valores son empíricos y se deben fijar tras una serie de pruebas con el algoritmo. En este caso se ha usado un valor de 15.

Un esquema del proceso de extracción de movimiento de los mapas de celdas se muestra en la Figura 14.

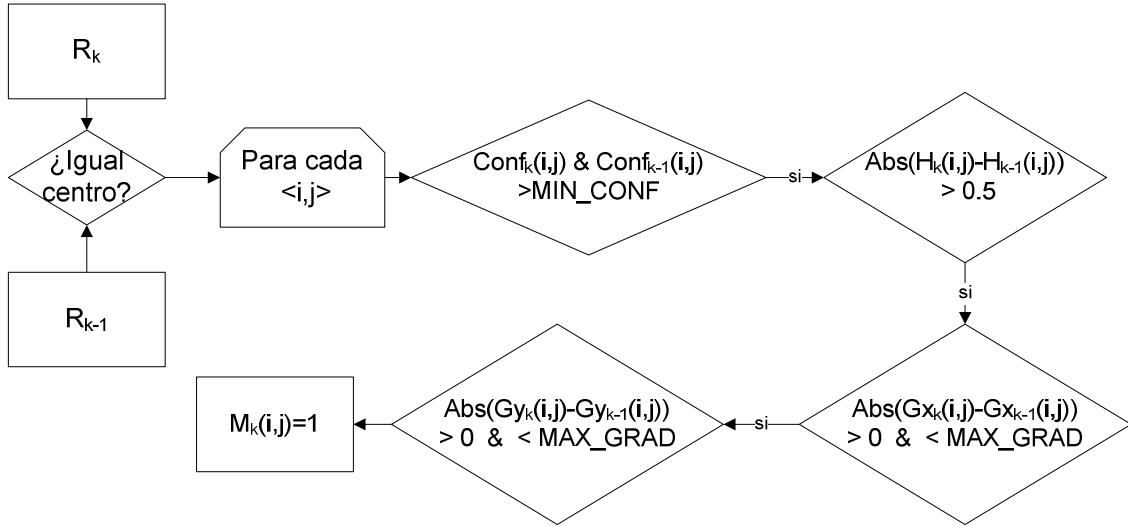


Figura 14: Esquema del algoritmo de extracción de movimiento

Cuando se dispone de dos mapas de movimiento consecutivos, obtenidos a partir de 3 reconstrucciones con el mismo centro, se combina la información de esos dos mapas para la obtención de las celdas donde se establecerá que se ha producido movimiento. Para ello se analiza cada una de las celdas del mapa de movimiento más reciente y se compara con la misma posición del otro mapa, incluyendo las celdas colindantes. Llamando M_k al mapa de movimiento más reciente y M_{k-1} al anterior, el proceso que se sigue para cada celda es el siguiente:

- Debe haber movimiento en M_k ; es decir, $M_k(i, j) = 1$.
- Debe haber movimiento en el entorno de la celda en M_{k-1} ; es decir, $M_{k-1}(u, v) = 1$ donde $[u, v] \in \{(i, j), (i + 1, j), (i + 1, j + 1), (i, j + 1), (i - 1, j + 1), (i - 1, j), (i - 1, j - 1), (i, j - 1), (i + 1, j - 1)\}$.
- Si se cumplen esas dos condiciones, $Mov(i, j) = 1$.

En la Figura 15 se observa un ejemplo de la obtención del mapa de movimiento. Para las celdas (2,2) y (5,3) existe movimiento en M_k y también hay movimiento en el entorno de la misma celda en el mapa M_{k-1} ; en concreto en las celdas (2,2) y (6,2). Sin embargo esto no ocurre para la celda (3,6), por lo que en el mapa de movimiento final (Mov), sólo se marcan las celdas (2,2) y (5,3).

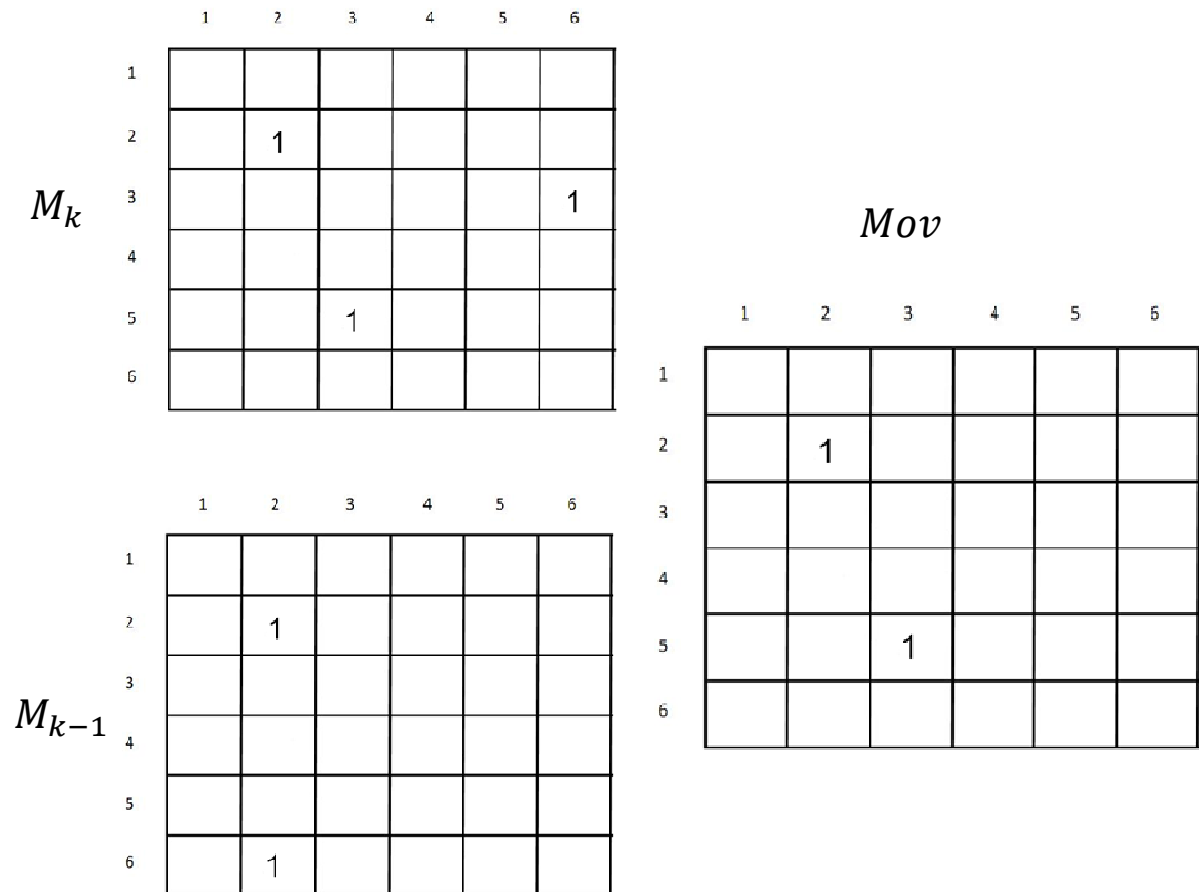


Figura 15: Ejemplo de obtención de mapa de movimiento

Una vez se termina de aplicar este algoritmo se obtiene un mapa de celdas de valores binarios en las que un 1 indica que se ha producido movimiento en esa celda. Este método no proporciona tanta información como los algoritmos desarrollados para el robot en posición estática, puesto que sólo indica las posiciones en el mapa donde se produce movimiento; pero no el tamaño, velocidad u otros datos del objeto móvil.

Con este mapa de celdas es posible obtener las posiciones reales en coordenadas globales donde se ha producido movimiento, permitiendo al sistema decidir si el robot debe detenerse y realizar un análisis más exhaustivo de esa zona o en qué dirección moverse para seguir a un posible intruso.

7. APLICACIÓN AL PROYECTO NM-RS. SIMULACIONES REALIZADAS.

Para comprobar el funcionamiento y la eficiencia de los algoritmos desarrollados se han realizado numerosas pruebas dentro del entorno de simulación creado en el proyecto NM-RS. De todas las pruebas realizadas se incluyen en esta memoria aquellas que se consideran más significativas para demostrar el funcionamiento de los algoritmos.

7.1. Detección de objetos desde robot estático.

7.1.1. Detección de un objeto en movimiento.

Descripción del escenario.

Esta primera prueba es la más simple; incluye un robot estático detectando un solo objeto móvil, representado por un robot en movimiento. La disposición inicial de los robots se puede observar en la Figura 16.

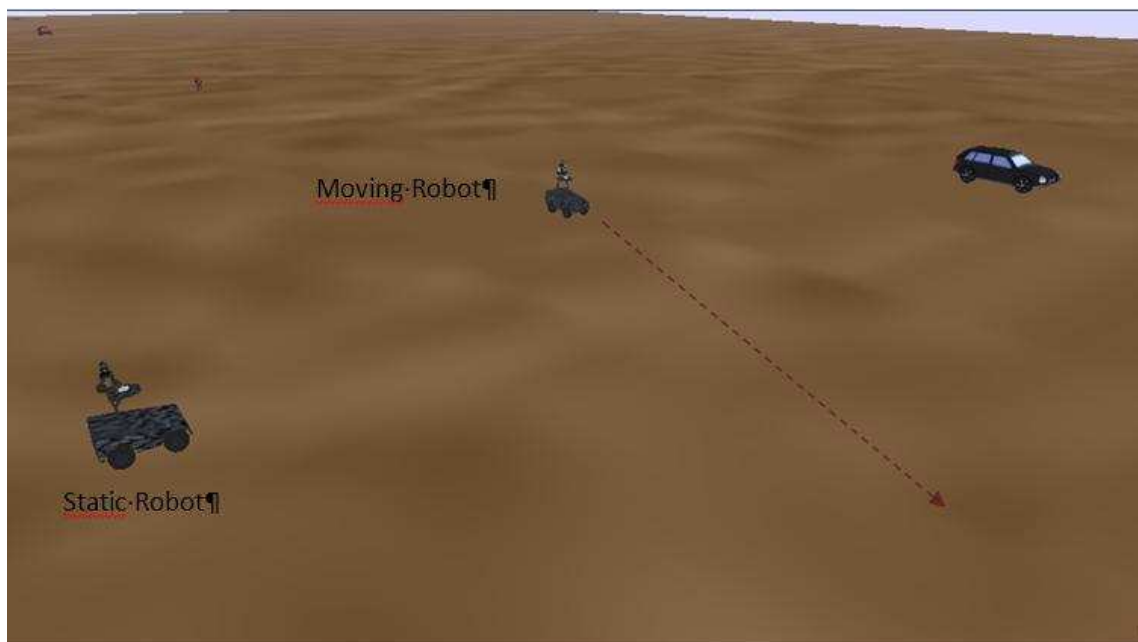


Figura 16: Disposición inicial de los robots

Objetivos de la prueba.

El principal objetivo de la prueba es comprobar el funcionamiento de los algoritmos de detección e identificación de objetos dinámicos. El sistema debería ser capaz de detectar un objeto en movimiento e identificarlo como el mismo objeto mientras permanece dentro del rango de detección del robot.

Descripción de los resultados.

En la Figura 17 se representa la trayectoria del robot móvil así como la posición de los objetos dinámicos detectados por el robot estático. También se presenta la posición del robot estático para poder entender cómo afecta la posición relativa de todos los elementos involucrados en el proceso en el resultado final.

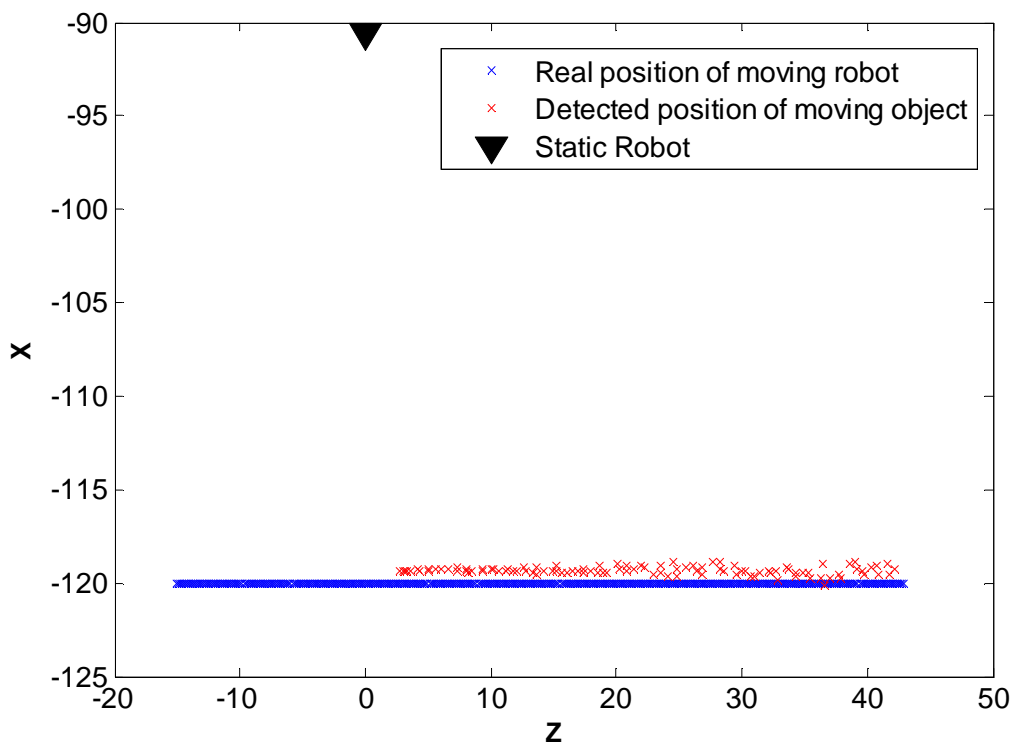


Figura 17: Resultados de la prueba con un sólo robot dinámico.

Análisis de resultados.

Como se observa en la Figura 17 el algoritmo de detección de objetos dinámicos funciona muy bien. Existe un periodo inicial en el que el algoritmo no se ejecuta porque la simulación tiene que estabilizarse, pero una vez comienza la detección de objetos dinámicos, el robot es detectado e identificado perfectamente a lo largo de su trayectoria. Sin embargo hay que clarificar dos aspectos de estos resultados.

En primer lugar, la posición real del robot en movimiento representada en la figura es la posición del centro del robot; el robot tiene alrededor de un metro y medio de ancho. Esto explica que la posición detectada está un poco desplazada hacia un lado porque el láser sólo detecta la cara del robot más cercana; es decir, la que tiene línea de visión directa con el sensor.

En segundo lugar, hay que reseñar que el rendimiento del algoritmo es menor a medida que el objeto se mueve más lejos del sensor. Esto se debe a que cuanto más lejos está el objeto, hay menos rayos del láser que colisionen con el robot y por lo tanto menos información que procesar.

7.1.2. Detección con oclusiones. Dos objetos en movimiento.

Descripción del escenario.

Este escenario incluye tres robots; uno estático que hace las veces de detector y dos dinámicos que hacen de objetos móviles. Los dos robots dinámicos se moverán desde dos posiciones diferentes en direcciones opuestas y se cruzarán en algún punto de su trayectoria. La Figura 18 muestra las posiciones iniciales para cada robot.

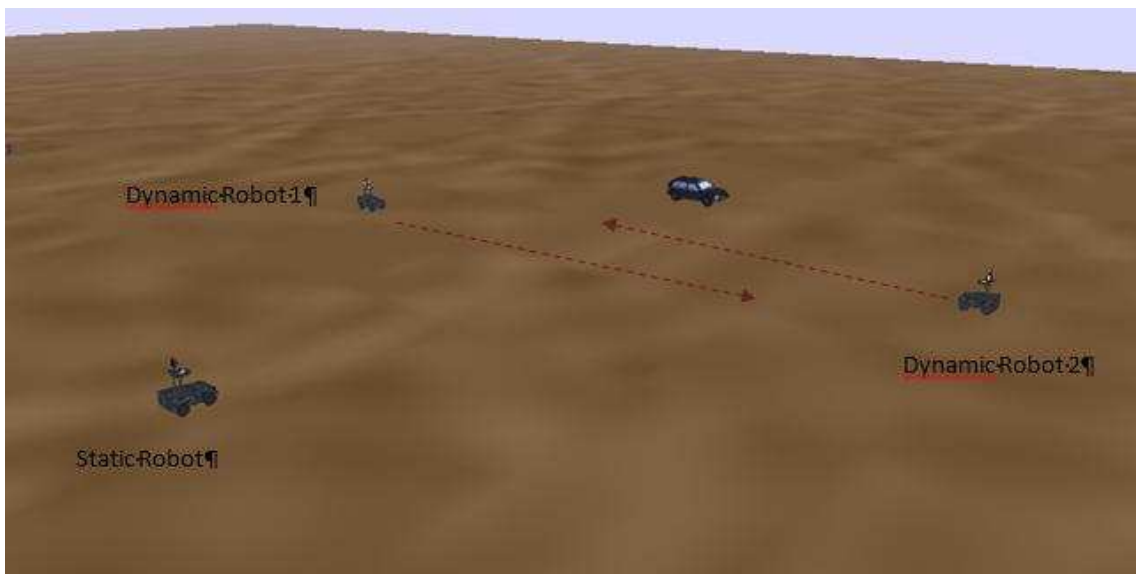


Figura 18: Posiciones iniciales de los robots

Objetivos de la prueba.

El principal objetivo de esta prueba es evaluar el rendimiento del algoritmo cuando se enfrenta con oclusiones. Como uno de los robots móviles ocultará al otro en algún momento de la prueba, el sistema debería ser capaz de identificar a los dos

robots correctamente e identificarlo cuando reaparezca después de haber estado oculto.

Descripción de los resultados.

Las trayectorias de los dos robots móviles se representan en unos ejes XZ; así como la posición del robot estático. Las trayectorias de objetos dinámicos detectadas también se representan en la misma gráfica. El robot estático sólo ha detectado dos objetos dinámicos en esta simulación. Los resultados se muestran en la Figura 19.

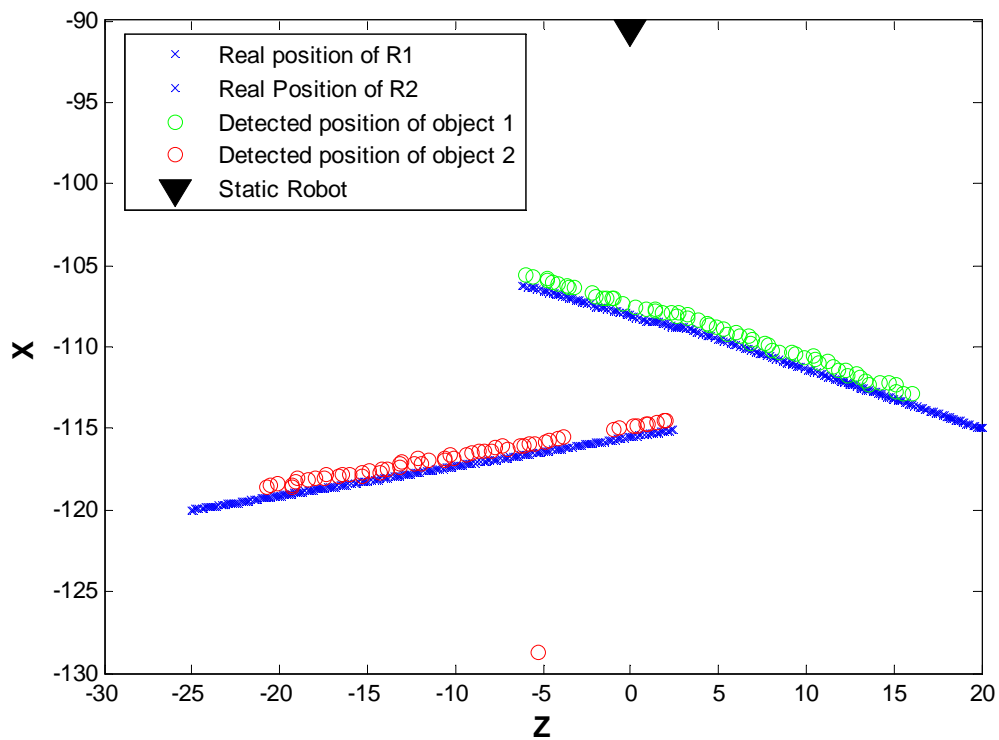


Figura 19: Resultados de la prueba con dos robots dinámicos.

Análisis de resultados.

El principal objetivo de esta prueba era comprobar el funcionamiento del algoritmo cuando se producen occlusiones. La gráfica muestra que el algoritmo desarrollado puede manejar occlusiones de forma efectiva. En la trayectoria del objeto detectado 2 se observa que hay un número de puntos que no aparecen detectados donde la trayectoria del primer robot bloquea el segundo robot de la línea de visión del

láser. Sin embargo, cuando el robot reaparece de nuevo todavía se lo identifica como el mismo objeto y su trayectoria se sigue de nuevo. Tan sólo existe un punto completamente erróneo en la trayectoria del objeto dinámico 2 y probablemente se debe a una identificación errónea durante la oclusión.

7.1.3. Prueba con 3 objetos

Descripción del escenario

En esta prueba intervienen cuatro robots, tres de ellos actuarán como objetos dinámicos y uno permanecerá quieto realizando la detección e identificación de objetos dinámicos. Las posiciones iniciales y las trayectorias de los robots se muestran en la Figura 20.

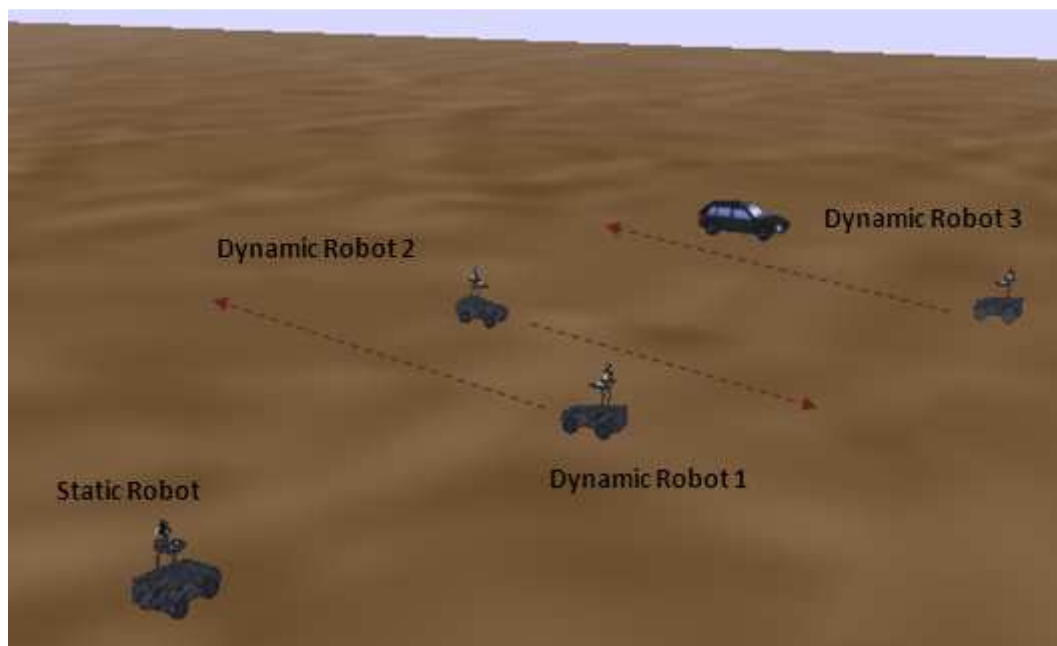


Figura 20: Posición inicial y trayectorias de los robots

Objetivos de la prueba

El principal objetivo de esta prueba es comprobar el funcionamiento del algoritmo cuando intervienen en el proceso más de dos robots. En este caso se han incluido tres robots que actuarán como objetos dinámicos. Una vez se comprueba que

el algoritmo puede manejar tres robots se puede asumir que puede manejar un número mayor de ellos.

Descripción de los resultados

En la Figura 21 se representan las trayectorias reales de los tres robots que actúan como objetos dinámicos. Sus posiciones iniciales son $(-100, -5)$ para el robot 1, que se mueve hacia la derecha; $(-110, 5)$ para el robot 2, que se mueve hacia la izquierda; y $(-120, -15)$ para el robot 3, que también se mueve hacia la derecha. También se ha incluido en los ejes la posición del robot estático para poder apreciar la distancia relativa entre el detector y los objetos móviles.

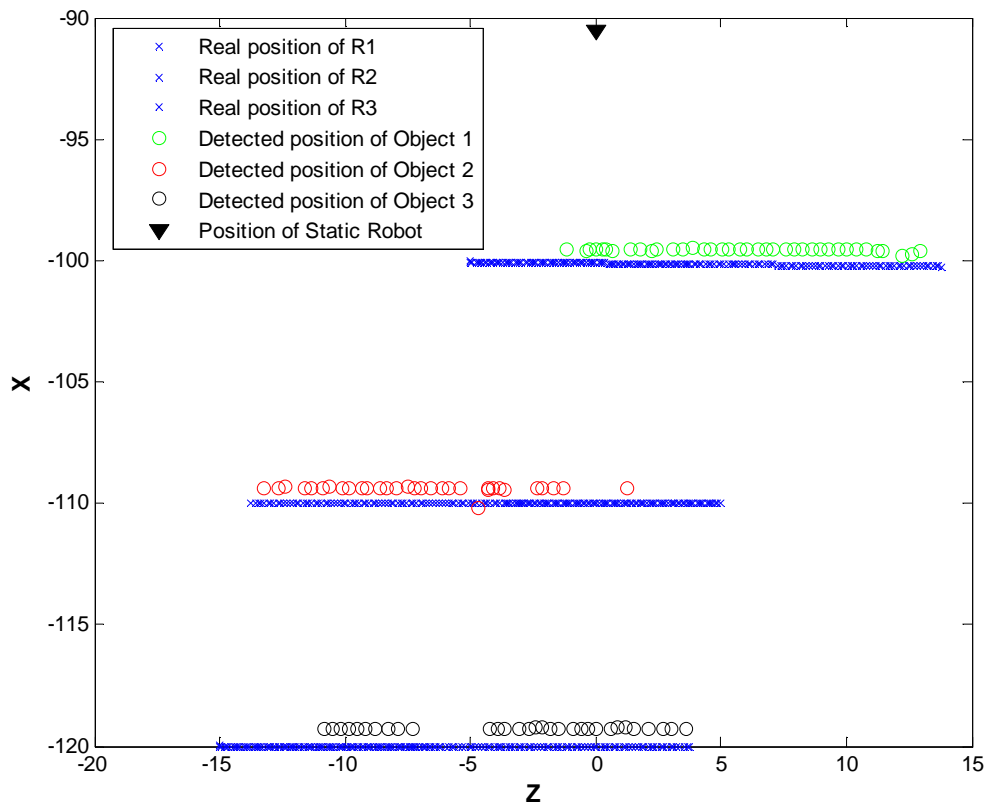


Figura 21: Resultados de la prueba con 3 robots

Análisis de resultados

Se ha comprobado que el algoritmo ha sido capaz de detectar y mantener identificados a tres objetos a lo largo de sus trayectorias. Tras el periodo inicial en el que el algoritmo se inicializa, y los objetos no aparecen en la lista, cada uno de los objetos es detectado correctamente. Además, los robots 2 y 3 son ocultos del robot estático por el robot 1 y en sus trayectorias detectadas se aprecian los dos huecos que

se producen con las oclusiones. En las dos ocasiones el algoritmo ha sido capaz de identificar correctamente a los objetos después de haber sido ocultados.

Por motivos de restricciones de hardware no se han podido realizar pruebas extensas con más de 3 robots ya que los equipos de los que se disponían no eran suficientemente potentes.

7.2. Detección e identificación con el algoritmo de filtro de Kalman

7.2.1. Detección e identificación de 3 objetos

Descripción del escenario

Esta prueba es la misma que la realizada con el algoritmo de mínima distancia; es decir, intervienen 4 robots, 3 como objetos dinámicos y 1 como detector. Las posiciones iniciales y trayectorias serán las mismas que las que se muestran en la Figura 20.

Objetivos de la prueba

Mediante esta prueba se pretende demostrar el funcionamiento del algoritmo de filtro de Kalman para la identificación de los objetos dinámicos. El proceso de detección es el mismo que el empleado con el algoritmo de mínima distancia, por eso se ha optado por incluir solamente la última prueba con los 3 objetos dinámicos. El algoritmo debería ser capaz de identificar los 3 objetos correctamente, incluso cuando queden ocultos por otros elementos del entorno.

Descripción de los resultados

En la Figura 22 se observan los resultados de las pruebas con el algoritmo de Kalman. Las trayectorias reales de los robots se representan en los mismos ejes junto con las posiciones de los objetos detectados. Las posiciones iniciales y trayectorias de los objetos son las mismas que en la prueba de 3 robots para el algoritmo de mínima distancia. En el gráfico también se incluye en el gráfico la posición del robot estático.

Para comparar el funcionamiento de este algoritmo de filtro de kalman (*K.F.A. Kalman Filter Algorithm*) con el algoritmo de mínima distancia (*M.D.A. Minimum Distance Algorithm*) también se incluye un gráfico con los dos resultados superpuestos en la Figura 23.

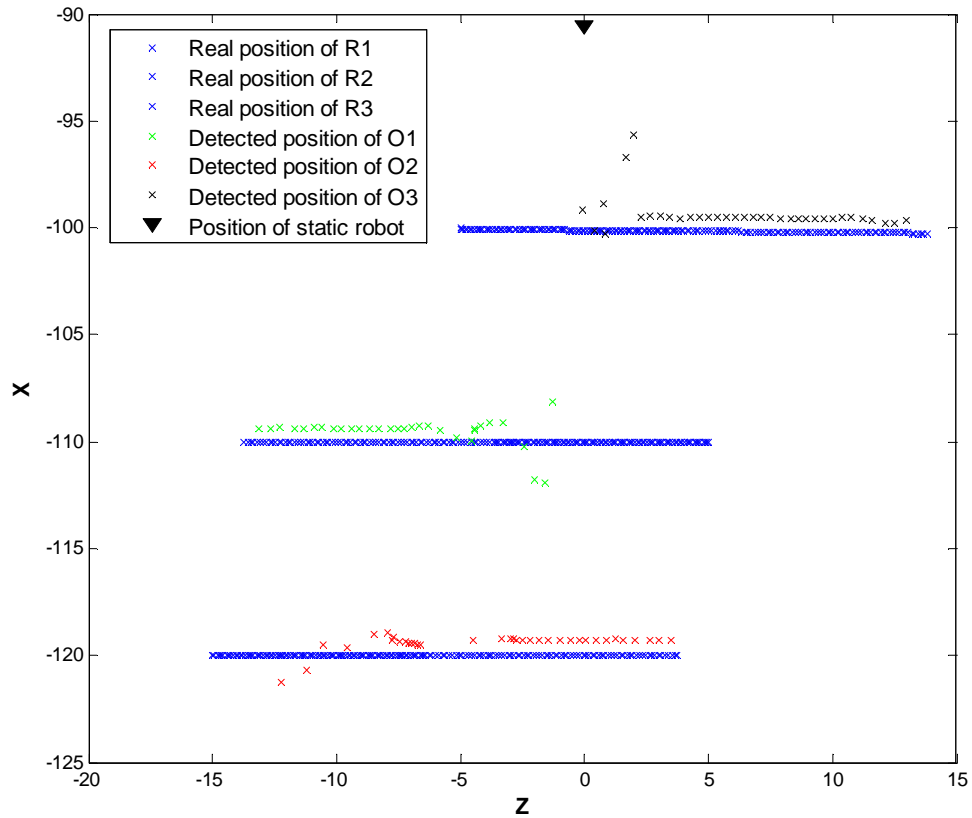


Figura 22: Resultados de la identificación con el algoritmo de filtro de Kalman

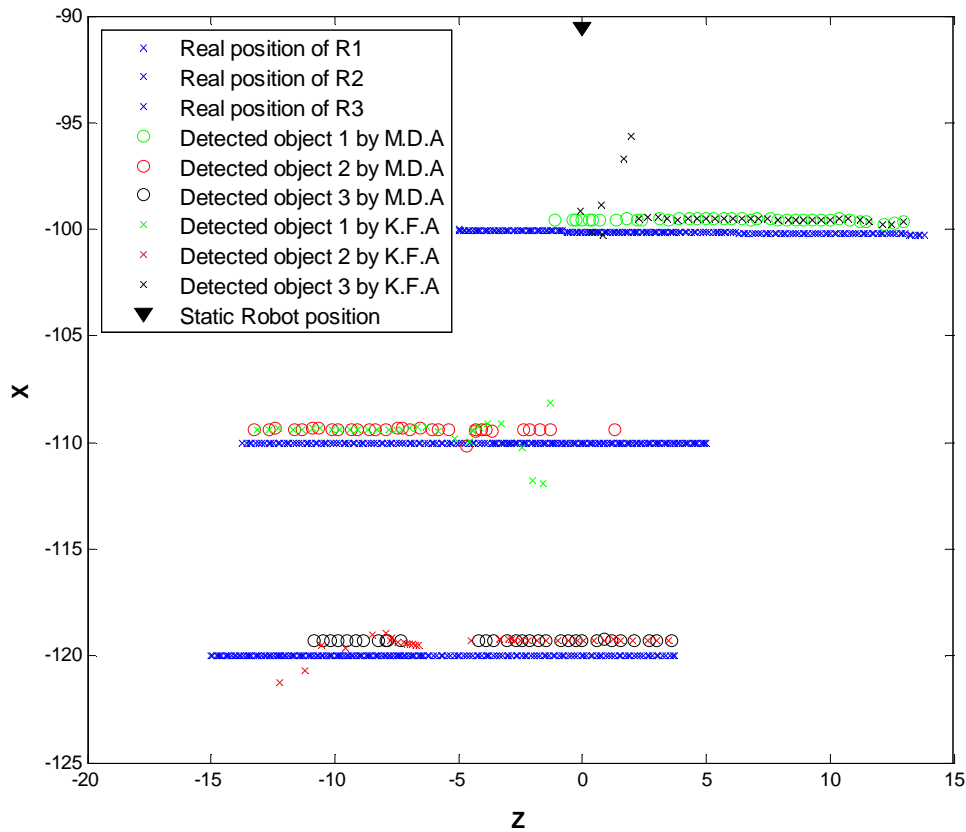


Figura 23: Resultados superpuestos de los dos algoritmos para la prueba con 3 objetos dinámicos

Análisis de resultados

En primer lugar hay que destacar que el algoritmo con filtro de Kalman tiene la misma efectividad que el algoritmo de mínima distancia cuando sólo se utiliza la información proveniente del láser. En esta simulación no se ha hecho uso de la información proveniente de la cámara o de la cámara termográfica, puesto que ambos robots tienen el mismo color y la misma temperatura.

En segundo lugar hay que señalar que el algoritmo de filtro de Kalman tiene un tiempo de inicialización más alto que el algoritmo de mínima distancia. Como todos los filtros de Kalman se necesita un tiempo inicial para que el filtro se estabilice. En la Figura 22 se observa como las trayectorias de los objetos detectados oscilan mucho al principio hasta que el filtro ha realizado varias iteraciones

En cuanto a las oclusiones, las funciones predictivas del algoritmo de filtro de kalman permiten extrapolar la posición de los objetos durante las oclusiones. Esta característica del algoritmo se puede observar en la Figura 23, en las trayectorias detectadas para el robot 3, en la zona donde se produce la oclusión. Se puede observar como la trayectoria detectada del robot 3 por el algoritmo de mínima distancia

acaba ligeramente antes que la trayectoria detectada por el algoritmo de filtro de Kalman al llegar a la oclusión.

7.3. Fusión de información entre varios robots

El proyecto NM-RS, la plataforma sobre la que se están realizando estas simulaciones, es un proyecto europeo en el que intervienen grupos de investigación de diferentes países. Aunque colaborar en un proyecto con tantos grupos puede ser muy enriquecedor; también presente unas dificultades añadidas, la más importante de las cuales es que se pierde el control de el conjunto del proyecto en beneficio de poder llevar a cabo tareas más grandes y complejas.

Esto ha influido en que no se hayan podido realizar las pruebas de fusión de información correctamente y no se hayan podido comparar las dos metodologías de fusión de información: centralizada y descentralizada.

7.4. Detección de movimiento desde robot dinámico

7.4.1. Dos robots enfrentados

Descripción del escenario

En este caso sólo se incluyen en el escenario dos robots dinámicos que se moverán en dos trayectorias paralelas cada uno en la dirección del otro. La posición inicial de los robots se observa en la Figura 24.

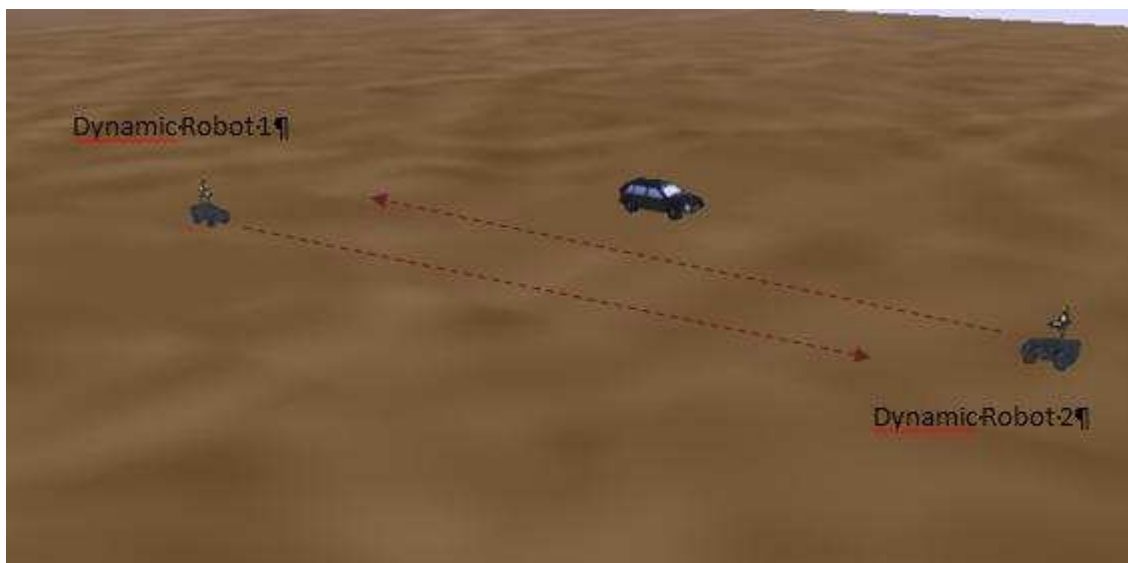


Figura 24: Posición inicial de los robots en la prueba

Objetivos de la prueba.

El principal objetivo de esta prueba es evaluar el rendimiento del algoritmo de detección de movimiento desde un robot móvil. Cuando un robot móvil detecta movimiento debe ser capaz de identificar la zona del mapa donde se ha producido este movimiento. En este caso, como los dos robots se mueven, uno de ellos actuará como objeto móvil (el algoritmo de detección de movimiento quedará desactivado) y el otro actuará de detector e intentará determinar si hay movimiento.

Descripción de los resultados.

Se representan en los mismos ejes las posiciones de los dos robots, tanto la del robot que ejerce de detector como la del que ejerce de objeto móvil. Sus trayectorias son paralelas y en sentido contrario, de forma que en un momento se cruzan entre ellos. Las posiciones iniciales de los robots son (-115,-15) para el robot 1, que se mueve hacia la derecha y (-120, 15) para el robot 2 que se mueve hacia la izquierda. Además se incluye en los mismos ejes las posiciones en las que el primer robot ha detectado movimiento. Todos estos resultados se representan en la Figura 25.

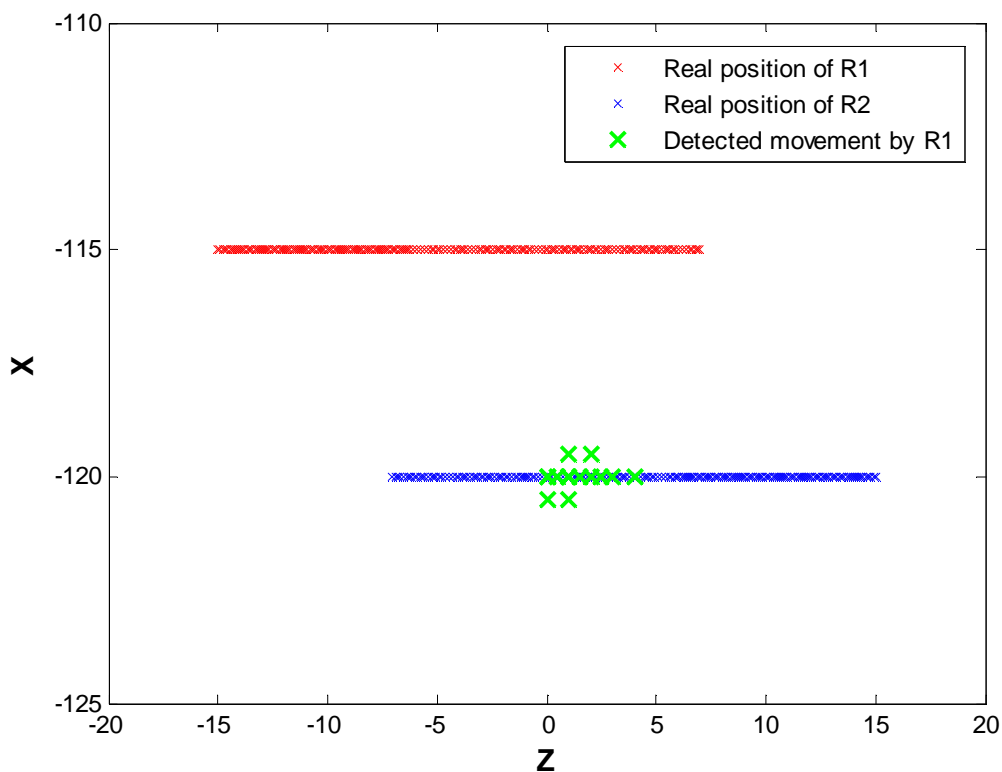


Figura 25: Resultados de la prueba de dos robots enfrentados

Análisis de resultados

En la Figura 25 se observa que el robot 1 ha detectado movimiento a lo largo de una parte de la trayectoria del robot 2. De estos puntos del mapa en los que se ha detectado movimiento hay que destacar dos aspectos. En primer lugar, se observa que los puntos detectados se encuentran cuando el robot ya ha avanzado una distancia considerable. Esto podría interpretarse como un fallo del algoritmo, aunque en realidad es una característica incluida para evitar que el robot detecte movimiento donde no lo hay. Las condiciones de detección de movimiento son muy estrictas, por eso sólo se detecta el movimiento en el entorno muy cercano del robot. En los puntos lejanos el movimiento se detecta con menor precisión y por eso no se marca, puesto que podría deberse a otras causas, como por ejemplo el cambio de altura debido a la detección progresiva de un muro.

En segundo lugar, a partir de cierto punto de la trayectoria se deja de detectar movimiento, debido a que los robots ya se han cruzado. Una vez que se cruzan el sensor deja de poder ver al otro robot, por lo que ya no se marca movimiento.

7.4.2. Dos robots en paralelo

Descripción del escenario.

En este caso sólo se incluyen en el escenario dos robots dinámicos que se moverán en dos trayectorias paralelas. Como se observa en la Figura 26 uno de los robots inicia su movimiento un poco más adelantado que el otro, de forma que uno actuará de objeto y el otro será el encargado de detectar toda su trayectoria.

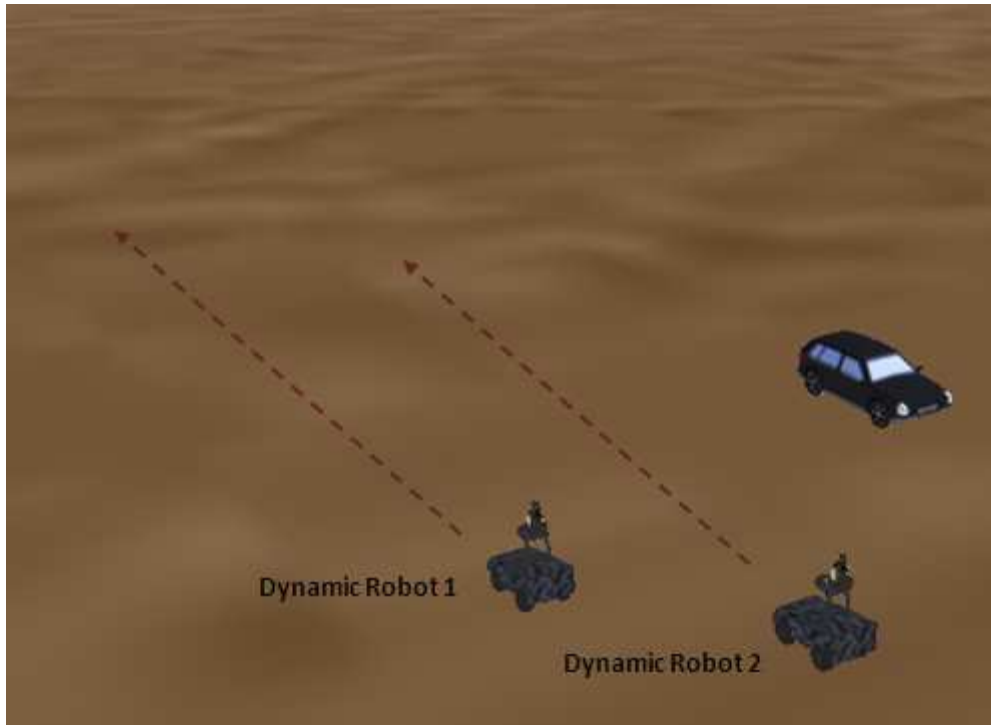


Figura 26: Posición inicial de los robots en la prueba

Objetivo de la prueba.

El principal objetivo de esta prueba es comprobar el funcionamiento del algoritmo de detección de movimiento cuando los dos robots se mueven en paralelo. Con este movimiento casi no habrá cambio en la imagen del láser debido al movimiento del robot; y, sin embargo, el algoritmo deberá detectar la zona por la que el robot se mueve.

Descripción de los resultados.

En la Figura 27 se observan las trayectorias de los dos robots móviles, cada una en un color distinto. Además se representan todos los puntos del mapa en los que el robot R2 ha detectado movimiento. Las posiciones iniciales de los robots son $(-117, -5)$ para el robot 1 y $(-121, -10)$ para el robot 2 y los dos se mueven hacia la derecha.

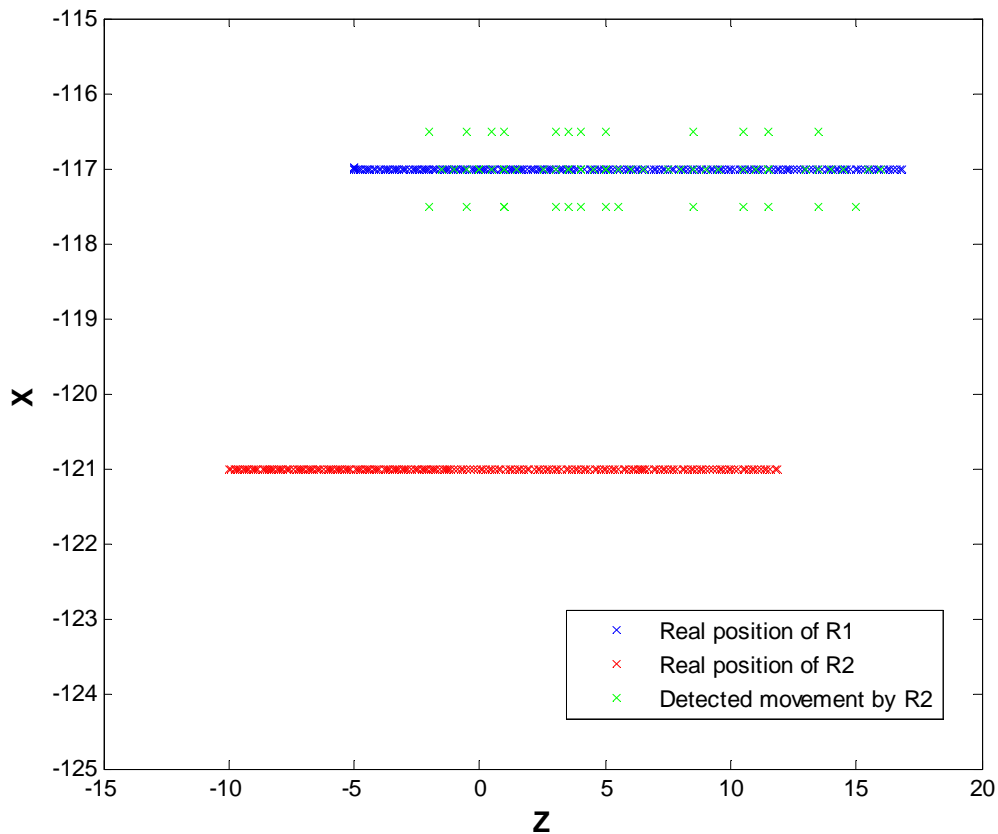


Figura 27: Resultados de la prueba de los dos robots paralelos

Análisis de resultados

En la gráfica de la Figura 27 se puede observar como el robot 2 ha detectado movimiento a lo largo de toda la trayectoria del robot 1. Los puntos detectados se encuentran sobre la posición del centro del robot y también a los lados, teniendo en cuenta que el ancho del robot es de más de 1m.

También se observa que, al desplazarse los dos robots a la misma velocidad, el robot 2 va detectando toda la trayectoria puesto que la distancia entre los robots es siempre la misma. Se demuestra por tanto que a pesar de no existir grandes cambios en la imagen del láser el algoritmo es capaz de detectar el movimiento correctamente.

8. LOGROS, CONCLUSIONES Y DESARROLLOS FUTUROS.

Revisando los objetivos de este proyecto, se puede afirmar que han sido cubiertos en su mayor parte. En cuanto al desarrollo, se han creado algoritmos para la detección e identificación de objetos dinámicos; así como un algoritmo de fusión de información entre diferentes robots que funciona en estructuras de control centralizadas y descentralizadas. En la validación de los algoritmos, sólo se ha conseguido comprobar completamente el funcionamiento de los algoritmos de detección e identificación de los objetos dinámicos. Al tratarse de un proyecto en el que colaboraban distintos grupos, parte del desarrollo no se realizaba dentro del grupo de investigación. En concreto, el envío de información entre los diferentes robots y con la base central no se había completado a la finalización de este documento. Esta es la principal razón por la que no se ha podido comparar la eficiencia del algoritmo de fusión de información en las dos modalidades para las que ha sido diseñado, el control centralizado y descentralizado.

Durante el desarrollo del proyecto se observó la necesidad de poder detectar las zonas del entorno en las que se producía movimiento mientras el robot se está moviendo, permitiendo así al robot decidir si debe detenerse para realizar una mejor detección e identificación de los objetos dinámicos en ese entorno. Para ello se desarrolló un algoritmo de detección de movimiento a partir de las imágenes del láser, cuyo funcionamiento también se comprobó exhaustivamente, incluyéndose en la memoria algunas de las pruebas realizadas.

Una vez evaluados los logros particulares de este proyecto, es conveniente comentar las conclusiones que se han sacado de la realización de un estudio de investigación de un campo concreto; en este caso la detección e identificación de objetos dinámicos.

En primer lugar hay que señalar la dificultad intrínseca que conlleva el trabajo con entornos y sensores tridimensionales en lugar de emplear con sensores en 2D. Una de las mayores dificultades del manejo de sensores 3D es la gran cantidad de información que hay que procesar. Por ello, una de las primeras conclusiones que se obtuvieron durante la realización del proyecto es que lo más conveniente es filtrar la información proveniente de los sensores antes de realizar algún procesamiento con ella. En este caso, para la detección de los objetos dinámicos se realiza una diferenciación de los escáneres, seguida de un filtrado para extraer los objetos móviles, antes de procesar estos puntos para la obtención de su posición real. En

muchos de los desarrollos con sensores 2D se realizan mapas de ocupación que representan todo el plano en el que se realiza un escáner, procedimiento que es inviable en entornos 3D debido a la gran cantidad de información que generarían.

Otro punto importante a destacar es que los datos provenientes del sensor láser son los más significativos a la hora de identificar un objeto. De los dos algoritmos desarrollados la posición de los objetos es el dato más útil que ayuda al seguimiento de los mismos mientras se mueven por el entorno. Mediante el algoritmo de filtro de Kalman se ha conseguido añadir a la identificación la información proveniente de otros sensores; en este caso la información de color proveniente de una cámara y la información de temperatura proveniente de una cámara termográfica; ayudando así a la identificación. Si bien la mejora en los resultados no ha sido muy grande, el método desarrollado puede servir para integrar la información proveniente de otros sensores que se añaden posteriormente a la plataforma.

Por último hay que señalar que se ha conseguido desarrollar un algoritmo simple para la fusión de información que funciona con diferentes estructuras de control, aunque no se ha podido comprobar su funcionamiento en pruebas reales debido a problemas en el desarrollo de otras partes del proyecto no dependientes de este grupo de investigación.

En cuanto a futuros desarrollos siguiendo esta línea de investigación, hay que señalar algunos aspectos que pueden ser mejorados en este tipo de sistemas. Uno de los mayores inconvenientes del sistema láser utilizados es el tiempo que tarda en realizar una medida. Aunque el tiempo que tarda el sistema LIDAR en realizar un escáner es muy pequeño, al tener que realizar muchos escáneres (uno por cada plano colineal al escáner) se incurre en un pequeño error puesto que no todos los puntos tienen el mismo tiempo asociado. Si bien este error es más significativo cuando el robot está en movimiento también puede afectar a los objetos detectados cuando el robot está estático; aumentando el error a medida que aumenta la velocidad relativa de los objetos implicados. Una posible solución es la utilización de escáneres LADAR, mencionados anteriormente en esta memoria, que obtienen una imagen tridimensional con medidas de distancia de forma instantánea.

Otra posible mejora a estos algoritmos proviene de la utilización de los otros sensores, no sólo para añadir información a los objetos detectados por el láser, sino utilizarlos también para la detección de objetos dinámicos. Con técnicas de visión artificial se podría utilizar la información proveniente de las cámaras para obtener listas de objetos dinámicos. Si bien los datos provenientes de las cámaras no incluyen información de distancia y por lo tanto no se podría obtener la posición real, si se

podrían obtener los puntos de las imágenes que pertenecen a cada objeto. Estos puntos podrían compararse con las listas de objetos obtenidas a partir de la información del láser para determinar si es posible que el láser haya pasado por alto algún objeto. Combinando estas dos listas se podría obtener una visión más acertada de la situación de los objetos dinámicos en el entorno del robot.

9. BIBLIOGRAFÍA

- [1]. www.robowatch.de
- [2]. www.technorobot.eu.es
- [3]. www.robotnik.es
- [4]. www.movironotics.com
- [5]. Agencia de Defensa Europea <http://www.eda.europa.eu>
- [6]. Documentación proyecto NM-RS.
- [7]. **Gao, T; Liu, ZG; Yue, SH.** *Traffic Video-based Moving Vehicle Detection and Tracking in the complex environment.* Cybernetics and Systems, 40 (7): 569-588 2009.
- [8]. **Watanabe, Y; Fabiani, P; Le Besnerais, G.** *Simultaneous Visual Target Tracking and Navigation in a GPS-Denied Environment.* IEEE 14th International Conference on Advanced Robotics, Vol. 1 and 2: 297-302 2009.
- [9]. **Hashimoto, M; Takahashi, K ; Matsui, Y.** *Moving-object tracking with multi-laser range sensors for mobile robot navigation.* IEEE International Conference on Robotics and Biomimetics, Vol. 1-5: 399-404 2007.
- [10]. **Hsia, SC; Hsiao, CH; Huang, CY.** *Single-object-based segmentation and coding technique for video surveillance system.* Journal of Electronic Imaging, 18 (3): Art. No. 033007 Jul-Sep 2009.
- [11] **Ahn, JH; Choi, C; Kwak, S; Kim, K; Byun, H.** *Human tracking and silhouette extraction for human-robot interaction systems.* Springer, Pattern Analysis and Applications, 12 (2): 167-177 Jun 2009.
- [12] <http://www.movirobotics.com/>
- [13] **A.Fernandez-Caballero, J.Mira, M.A.Fernandez, A.E.Delgado.** *On motion detection through a multi-layer neural network architecture.* NeuralNetworks 16 (2003) 205–222.
- [14] **B.C.Arrue, A.Ollero, J.R.Martinez de Dios.** *An intelligent system for false alarm reduction in infrared forest-fire detection.* IEEE Intelligent Systems 15 (3) (2000) 64–73.

- [15] **K.Tabb, N.Davey, R.Adams, S.George.** *The recognition and analysis of animated objects using neural networks and active contour models.* Neurocomputing 43 (2002) 145–172.
- [16] **J.B.Kim, H.J.Kim.** *Efficient region-based motion segmentation for a video monitoring system.* Pattern Recognition Letters 24 (2003) 113-128.
- [17] **J.Badenas, M.Bober, F.Pla.** *Segmenting traffic scenes from grey level and motion information.* Pattern Analysis and Applications 4 (2001) 28–38.
- [18] **S.Colantonio, M.Benvenuti, M.G.DiBono, G.Pieri, O.Salvetti.** *Object tracking in a stereo and infrared vision system.* Infrared Physics &Technology 49 (2007) 266–271.
- [19] **A.Fod, A.Howard, and M.J. Matari.** *Laser-based people tracking* in IEEE International Conference on Robotics and Automation (ICRA),(WashingtonD.C.), pp.3024–3029, May2002.
- [20] **Chen, Bafai; Cai, Zixing; Xiao, Zheng.** *Real-time Detection of Dynamic Obstacle Using Laser Radar.* 9th International Conference for Young Computer Scientists. IEEE 2008.
- [21] **M.Montemerlo, S.Thrun, and W.Whittaker.** *Conditional particle filters for simultaneous mobile robot localization and people-tracking* in IEEE International Conference on Robotics and Automation (ICRA), (Washington,D.C.), pp.695–701, May 2002.
- [22] **Yilmaz, A; Javed, O; Shah, Mubarak.** *Object Tracking: A Survey.* ACM Computing Surveys (CSUR) Article 13, Issue 4 2006.
- [23] **Joshua Bobruk and David Austin,** *Laser Motion Detection and Hypothesis Tracking from a Mobile Platform.* Australasian Conference on Robotics and Automation, 2004.
- [24] <http://en.wikipedia.org/wiki/File:LIDAR-scanned-SICK-LMS-animation.gif>
- [25] **Katz, R; Nieto, J; Nebot, E.** *Probabilistic Scheme for Laser Based Motion Detection.* Conference on Intelligent Robots and Systems, Pages 161-166 2008.
- [26] **Chieh-Chih Wang, Charles Thorpe, Arne Suppe.** *Ladar-based Detection and Tracking of Moving Objects from a Ground Vehicle at High Speeds.* Proceedings of the IEEE Intelligent Vehicles Symposium. 2003.

- [27] Iterative Closest Point. http://en.wikipedia.org/wiki/Iterative_Closest_Point
- [28] **Ashley W. Stroupe, Martin C. Martin, and Tucker Balch.** *Distributed Sensor Fusion for Object Position Estimation by Multi-Robot Systems*. Proceedings of the 2001 IEEE International Conference on Robotics & Automation. Seoul, Korea. May o 21-26, 2001.
- [29] **Manuel Mazo Jr., Alberto Speranzon, Karl H. Johansson.** *Multi-Robot tracking of a Moving Object Using Directional Sensors*. Proceedings of the International Conference on Robotics and Automation (ICRA) p. 1103—1108. 2004.
- [30] **Daniel Döhring, Hans-Dieter Burkhard.** *Multi Robot Object tracking and Self Localization using Visual Percept Relations*. Proceedings of the IEEE International Conference on Intelligent Robots and Systems. Octubre 2006. Beijing, China.
- [31] **S.Colantonio, M.Benvenuti, M.G.DiBono**, et. al. *Object tracking in a stereo and infrared vision system*. Infrared Physics & Technology 49 (2007) 266–271
- [32] **Jin-Xia Yu, Zi-Xing CAI, Zhuo-Hua Duan.** *Detection and tracking of Moving Object with a mobile robot using laser scanner*. Proceedings of the Seventh International Conference on Machine Learning and Cybernetics, Kunming, 12-15 July 2008.
- [33] Información filtro de Kalman. Wikipedia